UNIVERSITY
OF HULL

Addressing Complexity and Intelligence in Systems Dependability Evaluation

being a thesis submitted in fulfilment of the
requirements for the degree of
Doctor of Computer Science

in the University of Hull
by
Koorosh Aslansefat

Principal Supervisor:
Professor Yiannis Papadopoulos

Technical Expert:
Dr. David Parker

May, 2023

Declaration

I declare that the work presented in this thesis is the original work developed by the author. This work has been presented in international conferences and published in journals. The list outputs in the “Publications and Availability of Results” section show the relevance of these papers to chapters of the thesis, the scientific challenge that they address, and the key concept included in the paper. I would like to thank the co-authors of these papers for their contribution to the exposition and validation of this work.

Acknowledgements

This work would not have been possible without the support of people who supported me and shaped me as a better researcher.

I dedicate this thesis to my family who have always been my biggest support and encouragement: To my parents, Fatemeh Arabi and Rahman Aslansefat, without whose loving support this would not have happened. A heartfelt thanks to the love of my life, Mojgan Hashemian, for her unconditional patience and support all along.

Words cannot express my gratitude to my supervisor at the University of Hull, Professor Yiannis Papadopoulos for giving me the opportunity to carry out this work and for his valuable support and guidance throughout my PhD.

I would like to show my gratitude to Dr Martin Walker, and Dr David Parker, members of Dependable Intelligent Systems (DEIS) Lab., University of Hull, Dr Sohag Kabir from University of Bradford, Dr Youcef Gheraibia from York St. John University, Dr Ioannis Sorokos and Mohammed Naveed Akram from the Fraunhofer Institute for Experimental Software Engineering, and Dr Panagiota Nikolaou from KIOS Research and Innovation Centre of Excellence for sharing their pearls of wisdom and help me improve the concept.

I would like to thank Professor Athanasios Kolios from Department of Wind and Energy Systems, Technical University of Denmark, Glasgow, Dr Jose Ignacio Aizpurua from department of Electronics and Computer Science, Mondragon University. Their comments greatly improved the Butterfly Maintenance Model. Special thanks to Dr Gholamreza Latif-Shabgahi and Mehrdad Mohammadi for their comments on Markov Path theorem.

I would like to sincerely thank the financial support of my funders. EDF Energy R&D UK Centre, AURA Innovation Centre, and University of Hull, for providing the PhD Scholarship, the Alan Turing Institute and the Institution of Engineering and Technology (IET), for supporting my research and the European Union for their supported through the Secure and Safe Multi-Robot Systems (SESAME) H2020 Project under Grant Agreement 101017258.

Publications and Availability of Results

ID	Reference	Chapter	Challenge	Relation
1	Aslansefat, K., & Latif-Shabgahi, G. R. (2020). A hierarchical approach for dynamic fault trees solution through semi-Markov process. IEEE Transactions on Reliability, 69(3), 986-1003.	Chapter 2	Complexity	The Proposed Approach
2	Aslansefat, K., Kabir, S. Gheraibia, Y. & Papadopoulos, Y. (2020). Dynamic Fault Tree Analysis: State-of-the-Art in Modelling, Analysis and Tools, In Reliability Management and Engineering: Challenges and Future Trends, Taylor & Francis.	Chapter 2	Complexity	The Literature Review
3	Kabir, S., Aslansefat, K., Sorokos, I., Papadopoulos, Y., & Gheraibia, Y. (2019). A Conceptual Framework to Incorporate Complex Basic Events in HiP-HOPS. International Symposium on Model-Based Safety and Assessment, Lecture Notes in Computer Science 11842: 109-124, Springer, ISBN: 978-3-030-32871-9.	Chapter 2	Complexity	The idea of using Complex Basic Event
4	Aslansefat, K. & Papadopoulos, Y. (2019). DREAM: Data-driven Reliability-centred Evolutionary Automated Maintenance for Offshore Wind Farms, Global Offshore Wind 2019 – London.	Chapter 3	Complexity	The Proposed Butterfly Maintenance Model
5	Aslansefat, K. & Papadopoulos, Y. (2020). A Conceptual Framework for Data-driven Reliability-centred Evolutionary and Automated Maintenance of Offshore Wind Farms. EERA DeepWind'2020, Norway.	Chapter 3	Complexity	The Proposed Butterfly Maintenance Model
6	Aslansefat, K., Nikolaou, P., Walker, M., Akram, M. N., Sorokos, I., Reich, J., ... & Papadopoulos, Y. (2022). SafeDrones: Real-Time Reliability Evaluation of UAVs using Executable Digital Dependable Identities. In Model-Based Safety and Assessment. IMBSA 2022. Lecture Notes in Computer Science, Springer.	Chapter 4	Complexity	SafeDrones and Its Application

ID	Reference	Chapter	Challenge	Relation
7	Aslansefat, K., Kabir, S., Abdullatif, A., Vasudevan, V., & Papadopoulos, Y. (2021). Toward improving confidence in autonomous vehicle software: A study on traffic sign recognition systems. IEEE Computer, 54(8), 66-76.	Chapter 5	Intelligence	SafeML for Image Data

The following packages have been publicly provided in GitHub to increase the reproducibility of the research and help the other researchers to re-implement and use the provided functions.

Packages	Details	Related Chapters
	HDFT: Hierarchical Dynamic Fault Tree -- A Semi-Markov Process-based Hierarchical Solution for Dynamic Fault Trees. HDFT GitHub Repository: https://github.com/koo-ec/Hierarchical-Dynamic-Fault-Tree	Chapter 2, 3
	SafeDrones: A Framework for Reliability/Safety Modelling and Evaluation of Multi-copters (Multi-rotor Drones) and Electric powered Vertical Take Off and Landing (eVTOL) Aircrafts. SafeDrones GitHub Repository: https://github.com/koo-ec/SafeDrones	Chapter 4
	SafeML: Exploring Techniques for Safety Monitoring of Machine Learning Classifiers SafeML GitHub Repository: https://github.com/ISorokos/SafeML SafeML Extension for Time-Series: https://github.com/n-akram/TimeSeriesSafeML	Chapter 5

Awards and Honors

The following awards have been granted to the research presented in this thesis.

- ✓ IET Leslie H. Paddle Award for an Outstanding Researcher, 2020, (for the idea of Maintenance Modelling for Offshore Wind Farms).
- ✓ The Alan Turing Institute Post-Doctoral Enrichment Awards 2021, (for the idea of ML model monitoring using SafeML).
- ✓ Best paper Award for the paper entitled: "Combining Drone-based Monitoring and Machine Learning for Online Reliability Evaluation of Wind Turbines" at the IEEE International Conference on Computing, Electronics & Communications Engineering 2022 (IEEE iCCECE '22), University of Essex, Southend Campus, UK.
- ✓ Nominated for the Best paper Award for the paper entitled: "SafeML: Safety Monitoring of Machine Learning Classifiers through Statistical Difference Measures. In: Model-Based Safety and Assessment.", at the 7th International Symposium on Model-Based Safety and Assessment, Lisbon, Portugal.
- ✓ Best Poster Award for the paper entitled: "DREAM: Data-driven Reliability-centred Evolutionary Automated Maintenance for Offshore Wind Farms" at the Global Offshore Wind 2019, RenewableUK, London.
- ✓ Best PhD Project Poster Award in the Postgraduate Poster Competition at the School of Computer Science, University of Hull, 2023.

Abstract

Engineering and computing systems are increasingly complex, intelligent, and open adaptive. When it comes to the dependability evaluation of such systems, there are certain challenges posed by the characteristics of "complexity" and "intelligence". The first aspect of complexity is the dependability modelling of large systems with many interconnected components and dynamic behaviours such as Priority, Sequencing and Repairs. To address this, the thesis proposes a novel hierarchical solution to dynamic fault tree analysis using Semi-Markov Processes. A second aspect of complexity is the environmental conditions that may impact dependability and their modelling. For instance, weather and logistics can influence maintenance actions and hence dependability of an offshore wind farm. The thesis proposes a semi-Markov-based maintenance model called "Butterfly Maintenance Model (BMM)" to model this complexity and accommodate it in dependability evaluation. A third aspect of complexity is the open nature of system of systems like swarms of drones which makes complete design-time dependability analysis infeasible. To address this aspect, the thesis proposes a dynamic dependability evaluation method using Fault Trees and Markov-Models at runtime.

The challenge of "intelligence" arises because Machine Learning (ML) components do not exhibit programmed behaviour; their behaviour is learned from data. However, in traditional dependability analysis, systems are assumed to be programmed or designed. When a system has learned from data, then a distributional shift of operational data from training data may cause ML to behave incorrectly, e.g., misclassify objects. To address this, a new approach called SafeML is developed that uses statistical distance measures for monitoring the performance of ML against such distributional shifts. The thesis develops the proposed models, and evaluates them on case studies, highlighting improvements to the state-of-the-art, limitations and future work.

Table of Content

List of Figures

List of Tables

Nomenclature

ADD	Anderson-Darling Distance
BDD	Binary Decision Diagram
BE	Basic Event
CSP	Cold SPare gate
CVMD	Cramer-Von-Mises Distance
DBN	Dynamic Bayesian Network
DDEP	Demand Dependency
ECDF	Empirical Cumulative Distribution Function
EDDI	Executable Digital Dependability Identifier
FDEP	Functional Dependency
FPGA	Field Program Gate Array
FTA	Fault Tree Analysis
FTPP	Fault Tolerant Parallel Processor
GFT	Generalized Fault Tree
GSPN	Generalized Stochastic Petri Net
HARP	Hybrid Automated Reliability Predictor
HCAS	Hypothetical Cardiac Assist System
HCPS	Hypothetical Cascaded PAND System
HSP	Hot Spare Gate
IPC	Imperfect Coverage
KSD	Kolmogorov-Smirnov Distance
KD	Kuiper Distance
MAS	Mission Avoidance System
MCS	Minimal Cut Sequence Set
MDCS	Multi-processor Distributed Computing System
MDD	Multiple-Valued Decision Diagram
MDP	Markov Decision Process
MDPN	Markov Decision Petri Net
MTBF	Mean Time Between Failure
MTTF	Mean Time To Failure
MC	Markov Chain (Model)
NCSP	Cold SPare gate
NCSP	Hot SPare gate
NMR	N-Modular Redundancy
NVP	N-Version Programming
OBDD	Ordered Binary Decision Diagram
PAND	Priority AND
PFD	Probability of Failure on-Demand
RUL	Remaining Useful Life
SEQ	Sequential Enforcing Gate
TMR	Triple-Modular Redundancy
WD	Wasserstein Distance
WSP	Warm Spare Gate

Symbols

Symbols	Description
α	Dormancy Factor
λ	Failure Rate (FR)
λ_i	Failure Rate of Component *i*
α_s	Failure Rate of Warm Spare before of use
FS	Fail-Safe
FU	Fail-Unsafe
$L^{-1}\{\ \}$	Inverse of Laplace Stieltjes
$L\{\ \}$	Laplace Stieltjes
$F(t)$	Probability Distribution Function in Time Domain
$f(t)$	Probability Density Function in Time Domain
$f^*(s)$	Probability Distribution Function in Laplace Domain
c	Probability of a system fail-safe
C_i	Probability of fail-safe for component *i*
C_d	Probability of switch work correctly
P_i	Probability of State i^{th} in (Semi) Markov Model
$\hat{P}$	Probability Vector of Semi-Markov Model (LST)
$R(t)$	Reliability of a system at any time *t*
μ	Repair Rate
μ_i	Repair Rate of Component *i*
$S(t)$	Safety of a system at any time *t*
$G(t)$	Sojourn time Matrix in Semi-Markov model
$\tilde{g}(t)$	Sojourn time Matrix in Semi-Markov model (in LST)
S	States vector of Markov model
S_i	State i^{th} in Markov model
$Q(t)$	Expression: Unreliability of a system at any time *t* Matrix: Transient Matrix in Semi-Markov Model
$\tilde{q}(s)$	Transient Matrix in Semi-Markov Model (in LST)

Chapter 1

1 Introduction

Since the industrial revolution technology is growing exponentially, and it is expected to see even more rapid development in the future. Many technologies are used in safety critical applications where human life, or the environment are put at risk. Examples of safety critical systems include Air Traffic Control, surgical robots, chemical toxic processes and nuclear power plants. Technological disasters like the one seen in the Fukushima Daiichi nuclear power plant reminds us of the importance of building technologies that remain safe even in the most challenging circumstances (Perrow, 2011). The implication is that, when we invent new technologies or design new systems, safety evaluation should be a primary design concern rather than a secondary one and it should not be considered as an afterthought. Safety is an attribute of "dependability", an umbrella term that covers other attributes including reliability, availability, maintainability, and security. While the thesis focuses on safety it also addresses reliability and availability hence the term dependability is frequently used (Avizienis *et al.*, 2004).

1.1 Challenges and Motivation

This thesis is precisely motivated by a set of new challenges for dependability evaluation. A new generation of intelligent, autonomous, often open, and self-adaptive systems is emerging that poses difficulties for safety and dependability more broadly. Such systems include systems that employ machine learning or deploy swarms of collaborative autonomous robots. We identify two key challenges in the dependability evaluation of these emerging systems that, motivate this thesis – complexity and intelligence.

1.1.1 Challenge of Complexity

The complexity of a system often refers to the quality of being composed by wide range of interconnected parts or elements that exhibit a behaviour that is difficult to understand, model or predict especially in the context of their composition. It refers to the interconnectedness and interdependence of parts in a system, leading to behaviour that emerges from the interactions of these parts. Complexity can refer to a wide range of systems, including natural systems like ecosystems and weather, social systems like economies and cultures, and technological systems like computer networks. The study of complexity seeks to understand how these systems function. In addition, it seeks to understand and predict the behaviour of the systems with regards to timing. In general, complexity is characterized by a high degree of unpredictability, non-linear and dynamic relationships, and emergent behaviours.

1.1.1.1 Complex Architecture and Behaviour

In the perspective of safety science and dependability evaluation the same rules apply. Thus, it is hard to predict all the failures of a complex system, and dynamic behaviours such as priority, sequence, repair, and reconfiguration are difficult to be modelled. State-space models are well-known for handling the dynamic behaviour of systems. However, by increasing the number of components in a system, state-space models can face the issue of state-explosion in which the number of states in the system increases exponentially and the dependability model is no longer interpretable. Dynamic Fault Tree analysis is an approach for dependability modelling that is frequently used because of its intuitive nature and interpretability. But, for quantitative dependability evaluation it is not independent from state-space models. For having a quantitative reliability evaluation, Dynamic fault trees need to be transformed into other model types, such as Markov chains, Bayesian Network, and Petri Nets. However, this conversion

process can lead to a significant increase in the number of states, which is commonly known as the state explosion problem. This makes the analysis more complex and computationally challenging.

It is also limited to pre-defined dynamic gates such as SEQ, PAND, SPARE, and POR. A primary motivation of this thesis is to propose solutions that improve dependability modelling of complex architectures and behaviours whilst a) reducing state-space explosion b) increasing the model interpretability, and c) reducing its limitations with pre-defined dynamic gates.

1.1.1.2 Environmental Factors that Influence Dependability

System are not just machinery; the operate in physical and human environments. Environmental aspects can also make dependability evaluation more complex. For instance, the effect of environmental variables such as wind speed, wave hight and temperature can increase the complexity of understanding and modelling of an offshore wind farm. Creating a model that is both capable of modelling complex architecture and dynamic behaviours but also environmental factors in a challenge. The second motivation of this thesis is to propose solutions for that, in the context of dependability modelling in the domain of offshore wind. A model that deals with environmental factors should be capable of adapting itself based on the monitored conditions. So, we aim to provide a model that can update itself using real-time condition monitoring information and the estimated remaining useful life of the components of wind turbines.

1.1.1.3 Autonomy in Open Multi-Agent Systems

A system is autonomous when it decides on its actions, and it shapes its own configuration through its operational mission and life. Autonomy introduces unpredictability because decision algorithms can be highly complex. When autonomous systems fail, there is no operator to bring a system back into a safe state. Another form of unpredictability in emerging systems is caused by the open nature of loosely connected multi-agent systems like swarms of robots. These systems come together and dissolve in configurations that can be infinite in number and hard to predict at design time. Unpredictability is a major concern in dependability evaluation. Traditional techniques require all options and configurations to be enumerated and analysed at design time. However, this is not possible in a large class of emerging systems and the motivation is to find a solution to handle this kind of systems using the multi-UAV case study. A third motivation in this thesis is to propose solutions that address this problem for a multi-robot

drone system by using a recently proposed concept of Executable Digital Dependability Identities (EDDIs) (Aslansefat *et al.*, 2022). The goal is that dependability is continually evaluated during operation and the proposed mechanisms for that can then inform action and reconfiguration when dependability thresholds are violated.

1.1.2 Challenge of Intelligence

Many emerging systems include intelligent components, and this poses great difficulty in dependability evaluation. In classical analysis the system is assumed to be designed or programmed and there are models providing a basis of analysis. However, the behaviour of intelligent components is learned from data, and the component is typically a black box. Therefore, it is very hard to understand and predict the behaviour, and it is therefore also hard to predict at what conditions they might yield failure. There are several standards to evaluate traditional systems and software. One may model the failure behaviour of a function code using Fault Tree Analysis. However, there is no established standard for dependability analysis of a machine learning-based pedestrian detection algorithm. In this thesis the motivation with regards to the challenge of intelligence is to focus on distributional shift that can cause a machine learning algorithm to fail and use statistical distance measures to evaluate and monitor the shift.

1.2 Research Background and State-of-the-Art

The thesis deals with two different dependability evaluation challenges (complexity and intelligence) and three aspects of the challenge of complexity (complex architecture and behaviour, environment, and open nature of systems). Therefore, there are four aspects addressed. These are dealt with in four main chapters, and the background and the state-of-the-art is addressed individually for each aspect. Thus, for clarity, each chapter features its own distinct literature review, methodology, and presentation of numerical results and discussion. To set the scene, in this section, a brief background and state-of-the-art (SOTA) with respect to each aspect is provided. Details can be found in the respective chapters.

1.2.1 SOTA in Challenge of Complexity

In what follows, a brief background will be provided for each of the three aspects of complexity challenge.

1.2.1.1 SOTA in Complex Architecture and Behaviour

Safety-critical systems need to be designed with high reliability because system failures may contribute to hazards. Reliability is the probability of a system completing its expected function without any failure during its mission time (Dubrova, 2013). The accurate evaluation of reliability (and availability) when there is a dynamic behaviour such as active standby redundancy, dormant demand systems, aging of components, etc. is a challenging for dynamic systems as illustrated in Figure 1-1 (Norberg, Rosén and Lindhe, 2009).

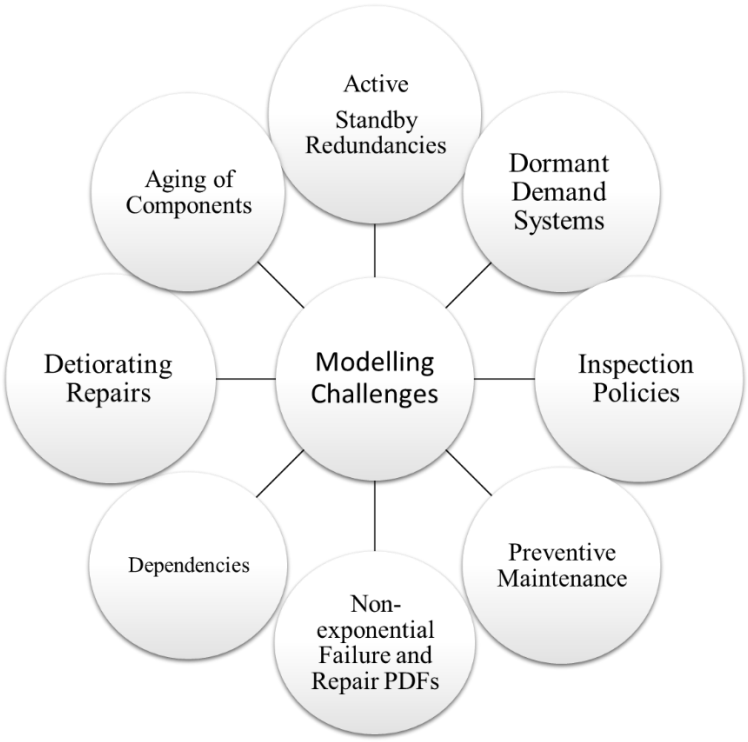


Figure 1-1. Modelling Challenges in Dependability (Norberg, Rosén and Lindhe, 2009)

Three main methods and theories are applied for reliability evaluation.

- State-space methods such as Continuous-Time Markov Chain (CTMC), Semi-Markov Process (SMP) and Markov Regenerative Process (MRGP) (Distefano, Longo and Trivedi, 2012b). In these methods, by increasing the number of system's components the model interpretability will be reduced and there is potential for state-explosion.
- Numerical methods like Monte Carlo and probabilistic expressions (Chiacchio *et al.*, 2013) are easier to perform but usually their answer is approximate, time-complex and its accuracy depends on the number of iterations in the algorithm.

- Combinatorial methods such as Reliability Block Diagram (RBD) and static Fault Tree (FT) (Misra, 2008). These are usually interpretable but limited for dynamic behaviour analysis. Static FT is a top-down graphical deductive technique, which is powerful in the description of systems' failures and their interactions. In addition, static FT has an independent reliability solution. However, the weakness of the static FT is in its inability to model the dynamic behaviour of systems such as functional and sequence dependence, spare and backup systems, priority, and repair. Dynamic Fault Tree (DFT) is introduced to model dynamic and time-dependent behaviour of systems through the novel gates introduced (Manian et al., 1999). Similar to static FT, DFT is powerful in the graphical representation of the system's failure interactions. Both Fault Tree (FT) and Dynamic Fault Tree (DFT) consist of different levels; top level and top event: usually, in the top level of a Fault Tree, there is a top event representing the failure of the whole system or mission. Intermediate level(s): this level includes the failure of sub-systems. As an example, the failure of an aircraft is a top event and the failure of its sub-systems such as the propulsion system, navigation system, etc. are the intermediate events located at an intermediate level. Basic events: in the FTA, a system can decompose to sub-systems and each sub-system can decompose to sub-sub-systems. This procedure will continue to the level that no more decomposition is affordable or possible. The events in the final decomposition level are called basic events. A failure of a GPS in a navigation system or a short circuit in an electronic board can be considered as examples of basic events. Gates: as mentioned before, the combination of failures in Fault Tree illustrates through logic gates (Kabir, 2017a).

In this thesis, DFT is used as a core model because of its high-interpretability feature. One solution is improved using a novel hierarchical Semi-Markov process (Aslansefat and Latif-Shabgahi, 2020). In addition, the complex basic event concept is also introduced to enhance DFT in dynamic systems modelling (Kabir, Aslansefat, *et al.*, 2019).

1.2.1.2 SOTA in Environmental Factors that Influence Dependability

There are several existing methods in which environmental factors are considered as a variable in dependability evaluation models. For example, a three-state Markov chain availability model of offshore wind turbine considering accessibility problems has been proposed by (Huang *et al.*, 2017). In that model weather delay, logistic delay and shift

time limitation regarding corrective maintenance has been studied. The results of the model have also been compared with Monte Carlo simulation to be validated. Another example is a Petri Nets-based parametric availability model proposed by (Santos, Teixeira and Soares, 2015) in which failure, repair, logistic time, weather delay have been considered. The article also considered Weibull distribution for rotor failure. Moreover, a queuing model constructed by a multi-state Markov process has been proposed to evaluate the availability of offshore wind turbines concerning the accessibility issues such as weather and transfer delays (Huang *et al.*, 2016). The main limit of these addressed model is that they are usually limited to a specific component.

In this thesis, using the concept of complex basic event that is presented in chapter 2 and the butterfly maintenance model in chapter 3, a novel approach will be presented to combine these models for different components and evaluate the availability of the whole system in the offshore wind turbine use case.

An additional aspect of taking environmental state into account is that the model must update their state using condition monitoring data or any external signal like the estimation remaining useful life. This means that the dependability model must update itself during operation, i.e. become executable. In this area, (Hurdle, Bartlett and Andrews, 2007) used a non-coherent Fault Tree for the fault diagnosis of a water tank system. The limitations in this method were a need for consistency checks from observation points. Two years later, the approach has been updated by combining the FTA and Bayesian Belief networks in (Lampis and Andrews, 2009).

Cai et al. (Cai et al., 2015) proposed a new method for real-time reliability analysis through a combination of traditional Bayesian networks derived from root cause diagnosis and dynamic Bayesian networks. In fact, this study update's prior reliability knowledge of the system (failure distributions) via dynamic Bayesian networks. A subsea pipe ram BOP system has been addressed as a case study in this paper. (Askarian *et al.*, 2016) proposed a new method for fault diagnosis through a fusion of micro-macro data. In this paper, the FTA and Bayesian networks have been combined to gain the advantages of both prior probability distribution in FTA and real-time data in Bayesian networks. Remaining Useful Life (RUL) is a parameter usually estimated through Machine learning approaches (Sikorska, Hodkiewicz and Ma, 2011).

A method for combining failure rate and RUL as the basic event in Dynamic Fault Tree has been proposed by (Aizpurua *et al.*, 2017). A hierarchical Bayesian network-based model has been provided for process monitoring and decision making by (Chen

and Ge, 2018). This article used a data-driven algorithm to update the sub-Bayesian networks in the model. (Getir *et al.*, 2018) has focused on semi-automated and co-evaluated process as a case study and defined a number of intra- and inter-model rules of transformation to cover the evaluation scenarios. The outcome of this study has shown that realizing the co-evolution of the proposed approach required fewer user interactions.

The potential challenges and opportunities of using machine learning in a safety-critical application have been reviewed in (Agrell *et al.*, 2018). The paper illustrated how missing casualties in the model can be reduced through the incorporation of safety models and data-driven knowledge. A conceptual idea regarding the combination of artificial intelligence methods with safety models has been presented in (Simen *et al.*, 2018). In this report, examples of golf-shot on the moon and Falcon launch from SpaceX have been demonstrated. (Cheng *et al.*, 2019) has proposed an Imitation Medical Diagnosis Method (IMDM) in which three types of Bayesian networks have been used: Machine Learning BN, Expert empirical BN, and maintenance decision BN. The method also applied the fuzzy theorem to achieve uncertainties and conditional probabilities.

Being able to link a dependability model with an external variable as a symptom is a vital step towards having an executable dependability model. In the next sub-section, the idea of executable digital dependable identifier (EDDI) with be discussed.

1.2.1.3 SOTA in Unpredictability in Open Multi-agent Systems

Despite existing guidance in terms of standards and guidelines there is significant diversity in the way how assurance of dependability properties is realized and demonstrated for open multi-agent systems. This generally makes it hard for third parties (e.g. certification authorities) to understand and assess the assurance approach, but particularly when systems are to be integrated (e,g, along a supply chain), this is a huge challenge which can consume significant amounts of time and money and also can lead to mistakes, ultimately jeopardizing the assurance of important system properties (Wei *et al.*, 2018).

Digital Dependability Identities (DDI) have been developed to address this problem. A DDI is modular, model-based and standardized. It contains all information relevant to describe the dependability characteristics of a system or component. At the heart of the DDI there is an assurance case, arguing the assurance of the relevant dependability properties and stringing together all models and artefacts (e.g. requirements, assumptions, architecture models, dependability analyses, evidences) that are required for the argumentation. A DDI is produced during design, issued when the

component or system is released, and is then continually maintained over the complete lifetime of a component or system. DDIs are used for the integration of components to systems during development as well as for the dynamic integration of systems to "systems of systems" in the field (Armengaud, D. Schneider, *et al.*, 2021) (Schneider *et al.*, 2015b).

Executable digital dependable identifier (EDDI) is an executable version of DDI in which the dependability artifact(s) can be executed inside a system (e.g. a robot) and also can communicate with other systems (other robots). As the idea is so novel there are few related articles about EDDI in the literature. On implementation of EDDI is based on conditional safety certificates (Schneider *et al.*, 2015b; Armengaud, D Schneider, *et al.*, 2021).

In this thesis we propose a novel use of Markov Models as executable EDDIs. In chapter 4, the idea behind this EDDI is developed and evaluated in a UAV-based case study (Aslansefat *et al.*, 2022).

1.2.2 SOTA in Challenge of Intelligence

Machine Learning (ML) is expanding rapidly in numerous applications. In parallel with this rapid growth, the expansion of ML towards dependability-critical applications raises societal concern regarding the reliability and safety assurance of ML. To address concerns, organizations and governmental institutes are working on new rules, regulations and standards for ML, such as in (ISO, 2017; Alexander *et al.*, 2020; on Standards in Public Life, 2020).

While ML is a powerful tool for enabling data-driven applications, its unfettered use can pose risks to financial stability, privacy, the environment and in some domains even life. Poor application of ML is typically characterized by poor design, misspecification of the objective functions, implementation errors, choosing the wrong learning process, or using poor or non-comprehensive datasets for training. Thus, safety for ML can be defined as a set of actions to prevent any harm to humanity by ML failures or misuse.

There are many perspectives and directions to be defined for ML Safety. (Amodei *et al.*, 2016) have addressed different research problems of certifying ML systems operating in the field. They have categorized safety issues into five categories: a) safe exploration, b) robustness to distributional shift, c) avoiding negative side effects, d) avoiding “reward hacking” and “wire heading”, e) scalable oversight as shown in Figure 1-2.

This categorization is helpful for an adequate assessment of the applicability a concept for a given (safety) problem. In the work presented here, we will be focusing on addressing distributional shift, however using a non-standard interpretation. Distributional shift is usually interpreted as the gradual deviation of the initial state of learning of an ML component and its ongoing state as it performs online learning. As will be shown later, distributional shift will instead be used by our approach to evaluate the distance between the training and observed data of an ML component.

Figure 1-2. AI Safety Evaluation Challenges (Amodei *et al.*, 2016)

It is also worth to note that there are different existing approaches for increasing the safety and robustness of ML algorithms. Some papers investigate the uncertainty evaluation of results in a classifier while others focus on the improvement of robustness against uncertainties. As an example, the following figure shows the ETH Robustness Analyzer for Neural Networks (ERAN) that uses possible perturbations for input "8" and tries to create a shape that abstracts all possible outputs. If the created shape violates the defined boundary and the results cannot be certified. Otherwise, the outputs will be guaranteed (Singh *et al.*, 2018).

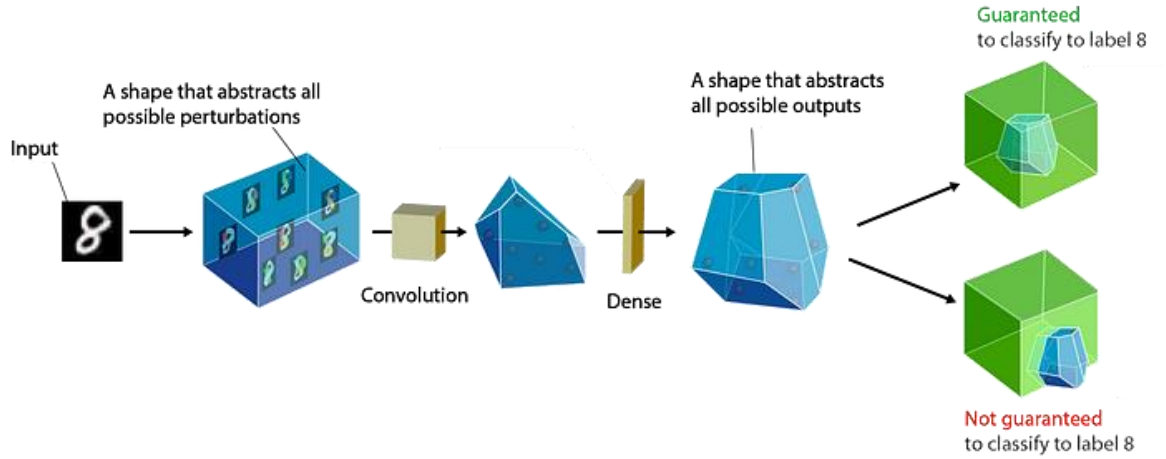


Figure 1-3. ETH Robustness Analyzer for Neural Networks (ERAN) (Singh *et al.*, 2018)

In this thesis, we explore the use of statistical distance measures for distributional shift monitoring in a new method that we call SafeML which relates such shifts to confidence in the reasoning of ML algorithms.

Statistical distance measures have been used to measure distributional shift. In modern ML algorithms like Generative Adversarial Nets (GANs), statistical distance or divergence measures are applied as a loss function, such as the Jensen-Shannon divergence (Goodfellow *et al.*, 2014), the Wasserstein distance (Gulrajani *et al.*, 2017), and the Cramer distance (Bellemare *et al.*, 2017). For dimension reduction, the t-SNE (t-distributed stochastic neighbour embedding) algorithm uses the Kullback-Leibler divergence as a loss function (Laurens van der Maaten, 2014). However, there is a gap for using the statistical distance measure for the purpose of safety monitoring and the aim is address it in chapter 5 for challenge of intelligence.

1.3 Research Context and Earlier Work on Challenges

This section gives some background on the process that led to this thesis. The work was originally funded through a collaboration with EDF Energy R&D in London, and the main focus was to develop a novel approach to provide a Data-driven Reliability-centred modelling for Offshore Wind Farms (OWF) which addressed the attributes of complex architecture and behaviour and environmental factors that influences dependability in the challenge of complexity.

During the first and the second year of the PhD, a novel approach for reliability/availability evaluation of complex systems is proposed. In addition, a new idea for connecting the monitoring data to the reliability/availability models was introduced to

make them data driven. At the end of the second year (2020), the IET Leslie H. Paddle has been awarded to the project to recognize its outstanding research outcomes.

In the third year, a new approach for safety monitoring of intelligent components in the system like Deep Learning/Machine Learning algorithms is proposed and it was part of addressing the challenge of intelligence. The method called **SafeML,** and it can provide an estimation of system performance in the presence of distributional shift and concept drift. SafeML provides a safety monitoring of such system through using statistical distances and proposing a human-in-the-loop procedure. It should be noted that the Alan Turing Institute enrichment award was granted for the expansion of the idea of SafeML in 2022.

In the write-up year, another research collaboration started in the European Commission Horizon 2020 project SESAME: Secure and Safe Multi-robot systems. The method described in this thesis has been used in this project for implementation of Safety-Targeted Executable Digital Dependable Identities (EDDIs) in multi-robot systems. A novel fault tree with symptom layer was proposed as a prototype for EDDIs to make the FTA data-driven and able to update the reliability profile of a system at runtime. The proposed model was used for evaluation the reliability of drones at runtime and a python package developed known as **SafeDrones**. Moreover, the SafeML idea was also used to monitor the safety of object detection (mostly human detection) algorithms used in robots.

1.4 Research Hypothesis and Objectives

The hypothesis of the proposed work is to the introduce a set of new models presented in the thesis that can partly address two challenges of complexity and intelligence with fours aspects of 1) Complex architecture and behaviours, 2) Environmental factors that influence the dependability, 3) Unpredictability in open multi-agent systems and 4) dependability vs. distributional shift.

To test this hypothesis for each aspect, the following objectives will be addressed.

- Research on relevant techniques to help inform a novel method that can meet the goals set in the hypothesis. The literature review is provided for each aspect:
    - **Complex Architecture and Behaviour**: Focusing on Dynamic Fault Tree as a model of choice and studying the existing solutions and their capabilities to address complex architecture and behaviour.

- **Environmental Factors that Influence Dependability**: Focusing on Offshore Wind Turbine Industry as a use case and investigating the literature for considering environmental factors in their proposed model.
- **Unpredictability in Open Multi-agent Systems:** Focusing on UAV case study and addressing the state-of-the-art for dependability evaluation of open multi-UAV systems.
- **Distributional Shift:** Investigating the existing research works on AI Safety evaluation.

- Proposing of New Models for each aspect:
  - **Complex Architecture and Behaviour**: Proposing a new hierarchical solution for DFT in order to handle complex architecture and behaviours.
  - **Environmental Factors that Influence Dependability**: Proposing a novel maintenance model for offshore wind turbines with the capabilities of considering environmental factors.
  - **Unpredictability in Open Multi-agent Systems:** Proposing an executable real-time reliability model for UAVs.
  - **Distributional Shift:** Using statistical distance measure as a mean to detect the distributional shift in machine learning applications and a proposed human-in-the-loop procedure to handle ML failure cases.
- Evaluation of the proposed methods using different use cases for each aspect:
  - **Complex Architecture and Behaviour**: Using the existing benchmarks for DFT evaluation as well as considering the use case aircraft fuel distribution system.
  - **Environmental Factors that Influence Dependability**: Considering offshore wind turbine and its availability evaluation as a use case for this aspect.
  - **Unpredictability in Open Multi-agent Systems:** Considering UAV case study and its real-time reliability evaluation.
  - **Distributional Shift:** Utilizing ML toy benchmarks as well as considering security intrusion detection benchmark.

1.5 Summary of Contributions

The main contribution of this PhD thesis is proposing a novel hierarchical approach for reliability and availability evaluation through combined dynamic fault tree and semi-

Markov process. Considering the proposed Hierarchical DFT (HDFT) solution as a backbone of the whole thesis, the following contributions are then added as extensions that can improve the capabilities of HDFT to handle complex behaviour, maintenance tasks, runtime reliability evaluation and dealing with intelligent systems. The list of all contributions has been summarized as follows:

1.5.1 Challenge of Complexity

1.5.1.1 Complex Architecture and Behaviour

- The proposed Hierarchical DFT (HDFT) solution is capable of handling non-exponential and complex failure behaviours thanks to Semi-Markov Process capability.
- Introducing the idea of complex basic events that increase the capability of dynamic fault tree analysis and push it beyond pre-defined dynamic gates.
- Improving the previous approach using the idea of Markov path that can break-down Markov and Semi-Markov models. It can reduce the number of states and transitions that leads to lower computation complexity and reduce the issue of state explosion in Markov models. In other words, it can reduce the state explosion issue in using state-space models like Markov.

1.5.1.2 Environmental Factors that Influence Dependability

- Proposing a novel method that considers both traditional and data-driven availability analysis by combining both simple failure distribution and remaining useful life (RUL)-based transition in a single MRGP. This is the first time that MRGP is combined with the real-time transition. The proposed model enables users to evaluate both condition-based maintenance and reliability-centered maintenance.
- The introduction of an integrated BMM model incorporating more advanced logistics delays, weather delays and a daily briefing and transfer delays. Consideration of these accessibility and delay factors will lead to a more realistic availability evaluation of the offshore wind farm.
- In the Markov modelling theories, the issue of considering the time-based deadline for a transition has rarely been discussed. This article uses the deadline transitions to model shift time limits of the O&M for offshore wind turbines.
- Imperfect maintenance can make the model and its evaluation more realistic. However, the literature lacks proper modelling for imperfect maintenance. The

proposed model is not only able to consider imperfect maintenance but also able to use this option to model the life extension of the turbine.

- The presented model can model non-exponential system behaviours using MRGP. The MRGP has higher computation complexity but at the same time offers more power in modelling complex failure behaviours. At the same time, the embedded and presented Markov model can be used with Markov processes in cases where only exponential failure and delay behaviours is considered. In other words, the computational complexity of the model is flexible and can be reduced via simplifying assumptions.

1.5.1.3 Unpredictability in Open Multi-agent Systems

- Combination of Dynamic Fault Tree Analysis with complex basic events and proposing symptom layer to support real-time reliability evaluation.
- Introducing the concept of executable model for multi-UAV systems as a prototype of the EDDI concept known as SafeDrones.
- Introducing a novel symptoms layer for the Dynamic Fault Tree to integrate it with runtime monitoring data.

1.5.2 Challenge of Intelligence

- Proposing SafeML approach for Safety monitoring of Deep Learning/Machine Learning algorithms using statistical distance measure and a human-in-the-loop procedure. SafeML is able to estimate the performance degradation of DL/ML algorithms in the presence of distributional shift and concept drift.
- SafeML is a framework that is easy to implement and can accommodate various distribution families, including exponential and normal distributions.
- Being a model-agnostic approach, SafeML can deal with variety of Machine learning and deep learning algorithms.
- A new human-in-the-loop procedure is proposed to enable real-time safety monitoring of machine learning algorithms. This procedure operates at three levels: I) Accepting a decision and providing a real-time estimate of its accuracy. II) Detecting when there is insufficient data available and buffering more samples to improve decision-making. III) Rejecting a decision and requesting the involvement of a human agent (In cases that reaction time is not a limit). This occurs when the algorithm is unable to make a reliable decision on its own.

1.6 Thesis Structure

The rest of this thesis is organized into two main sections (for challenge of complexity and challenge of intelligence) and consists of six chapters.

Challenge of Complexity

This section consists of three main chapters discussing challenges of complexity for quantitative safety analysis.

Chapter Two – Complex Architecture and Behaviour

This chapter starts with a discussion on how and why the Dynamic Fault Tree and Markov theorem is selected for reliability and availability evaluation. In addition, a comparison of different existing Markov models will be provided briefly. The Markov theorem and its mathematical equations is provided for both transient and steady-state evaluation. Moreover, the hierarchical Semi-Markov Process (SMP)-based solution for Dynamic Fault Tree is proposed and the concept of Markov path theorem is explained for the first time. Various examples are given to show the capabilities of the proposed method. The proposed method is used as a core for chapter three and four.

Chapter Three – Environmental Factors that Influence Dependability

This chapter starts with highlighting the importance of renewable energies and focusing on offshore wind energy. The challenge of external and environmental variables and their effect on availability models (as a subcategory of complexity challenge) is addressed in this chapter. In addition, this chapter provide an answer to the question of how the estimated remaining useful life can be included in the traditional quantitative safety models. Introducing the Butterfly Maintenance Model (BMM) is the main contribution of this chapter. In this chapter, the hierarchical dynamic fault tree analysis from chapter two is combined with the idea of complex basic events to handle BMM from different components of offshore wind turbines. Different examples are given to illustrate the capabilities of the proposed method. It should be mentioned that this part is related to my industrial collaboration with EDF Energy company.

Chapter Four – Unpredictability in Open Multi-agent Systems

In this chapter, another industry is chosen to showcase a different perspective of complexity challenge. The outcome of this chapter is the **SafeDrones** Python package as part of the EU SESAME Project. The general idea of this chapter is to show how to make the Safety model executable and how the safety models can be linked to the monitoring and diagnosis system. Moreover, a novel approach is proposed to use complex basic event

and Markov model and provide a real-time reliability evaluation of UAVs. In order to demonstrate the advantages of the proposed method, it has been implemented on a real industrial case study in collaboration with the KIOS Research and Innovation Centre of Excellence as one of industrial use case partner of the EU SESAME project.

Challenge of Intelligence

In this section the challenge of intelligence will be addressed.

Chapter Five – Safety Monitoring vs. Distributional Shift

The focus of this chapter is to highlight the importance of AI safety and its directions. In addition, distributional shift is discussed a one of the main reasons for performance drop in Machine Learning algorithms. In this chapter, SafeML is proposed as a solution for safety monitoring of machine learning classifiers. The chapter provides several examples to show how SafeML can be effective for machine learning monitoring at runtime. It should be mentioned that the idea won the Alan Turing Institute Award and in the EU SESMAE project there a number of industrial partners were interested to test the proposed method for their industrial use case.

Chapter Six – Conclusion

The summary of the thesis with an overall conclusion is provided in chapter six. In addition, the capabilities and limitations of the proposed ideas is discussed in this chapter and a couple of suggestions for future works are addressed.

Chapter 2

2 Addressing Complex Architecture and Behaviour in Dependability Evaluation

In this chapter, we tackle difficulties in dependability evaluation introduced by complex architecture and dynamic behaviour. A new method is proposed named as HDFT (Hierarchical DFT solution using Semi-Markov Process) that combines Dynamic Fault Tree and Markov models. The chapter provides a comparative review of different existing Markov models. The Markov theorem and its mathematical equations are provided for both transient and steady-state evaluation. Hierarchical Semi-Markov Processes (SMP) are integrated with Dynamic Fault Trees and the novel concept of a Markov path theorem is introduced. Numerical examples of application on systems show the capabilities of the method.

2.1 Dynamic Fault Tree Analysis

Over the years, several methodologies have been developed to facilitate safety analysis and more broadly dependability analysis that includes prediction of reliability and availability of systems. Among them, Fault Tree Analysis (FTA) is one of the oldest and most popular techniques widely used to perform safety and reliability analysis of systems. In traditional FTA, systems and their components are usually considered to have two states: working and failed. To model the logical interaction between different failure events Boolean AND and OR gates are used, and the causes of system failure are determined in the form of combinations of events. To facilitate reliability analysis, each of such component can have its probability of failure or failure rate or distribution of time of failure or steady-state or instantaneous unavailability/availability defined. At the same time, if the component can be repaired then a repair rate is defined. However, modern large scale complex systems have the capacity to work in different states and they can have a complex repair process. A component in such system can work as a primary component at a particular point in time, and in another time instance the same component can work as a secondary component. Moreover, if a component acts as a spare component in a system, it can be in a different mode of spare such as cold, warm, and hot spares.

Such multi-modal operation capability of systems and complex interactions between their components gives rise to different dynamic failure characteristics like priorities among events and functionally dependent events. However, using a classical fault tree approach it is not possible to explicitly consider system dynamics and sequencing/timing of events while performing analyses, which may produce inaccurate results (Kabir, 2017b). The limitations of the classical analysis techniques have not gone unnoticed, and it was recognised that methodologies with more powerful modelling capabilities are required to take into account the dynamic behaviour of systems for a comprehensive and accurate analysis of complex systems.

Several attempts have reported in the literature to improve the modelling power of SFTs through augmentation to include different types of temporal and statistical dependencies in the FT model. In 1976, the concept of Priority-AND (PAND) gate was introduced by (Fussell, Aber and Rahl, 1976). Later, several extensions to the SFTs such as the DFT (Dugan, Bavuso and Boyd, 1992; Dugan, Sullivan and Coppit, 2000), temporal fault trees (Palshikar, 2002; Walker, 2009), and State/event fault trees (Kaiser, Gramlich and Förster, 2007) have been proposed. Among these extensions, DFT is the most popular dynamic extension of SFTs. The DFT retains the PAND gate and

additionally, it introduces new dynamic gates like Functional Dependency (FDEP), SPARE and Sequence Enforcing (SEQ) gates.

Over the years, significant advancement has been made in the area of dynamic system analysis using DFTs. In this chapter, I reviewed different method of DFT analysis, which include both qualitative and quantitative analysis approaches for DFT analysis. Development in qualitative analysis started with the extension of the concept of minimal cut sets of SFTs to the minimal cut sequences (MCSQs) of DFTs. This was followed by the introduction of approaches for the determination of MCSQs from the structure of DFTs. On the other hand, the development in the quantitative analysis area mainly focuses on the quantitative evaluation of the top event of the DFT based on the quantitative failure behaviour related information, e.g., failure rate or probability of the basic events. To accomplish this task, a number of existing approaches such as Markov models have been addressed.

2.2 Overview of Dynamic Fault Tree Analysis

Dynamic Fault Tree has a similar logical structure to its static counterpart. The event at the top of the tree is known as the top event (TE), which almost always represents a system failure. This top event is decomposed into a combination of intermediate events (IE). Unlike the static fault tree, DFT uses both Boolean and dynamic gates to specify logical relationships among events to represent the IEs. IEs are further decomposed down to lowest-level events, which are known as basic events (BEs).

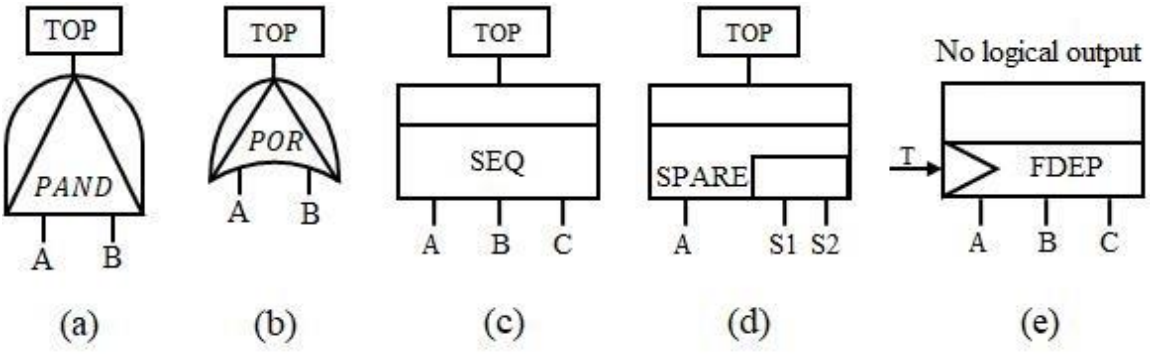


Figure 2-1. DFT Logic Gates

To allow the fault tree to model sequence/time dependent failure behaviour of systems, several dynamic gates have been introduced. Figure 2-1 shows the commonly used DFT gates. Priority-AND (PAND) gate is a special version of the AND gate. It delineates the priority behaviour in a dynamic system. In this gate, the output will be true

when both inputs occur, and the first input (event A) occurs sooner than the second input (event B). In other words, the occurrence time of event A should be less than the occurrence time of event B and both of them should fail to have the failure as the output of this gate. Like the PAND gate, the Priority-OR (POR) gate also delineates a sequence, however, it defines an ordered disjunction rather than an ordered conjunction. In this gate, first input (event A) has priority over other inputs. This event must happen first for the POR gate output to be true, but does not require all other events to occur(Walker, 2009). If other non-priority events occur, they must occur after the priority input. The Sequence-Enforcing gate (SEQ) gate represents the sequential failure behaviour of events A, B and C respectively. It means events B and C cannot fail before the failure of event A. Also, event C cannot fail before the failure of event B.

The SPARE gate is used to model redundancy in system design. The inputs to the SPARE are all BEs. The leftmost of the input corresponds to a primary event and other inputs represent spare components. In SPARE gate of Figure 2-1 (d), the input A is the primary component and S1 and S2 are two spare components. The behaviour of this gate is defined as such that when the primary component (A) fails the first spare (S1) will be activated; and if S1 fails then S2 will be activated. Finally, the outcome of the gate will become true when all of its inputs become true. A SPARE gate can represent three different types of dynamic redundancy; I) CSP: Cold Standby Spare in which the spare parts will be activated to be replaced when the primary unit (A) fails. That means in the cold spare mode the spare components are deactivated until they are required. II) HSP: Hot Standby Spare in which the spare part starts to work in parallel with primary unit and when it fails the spare part will be replaced immediately. III) WSP: Warm Standby Spare in which the spare part partially works in parallel with the primary unit to be replaced when needed. In other words, the spare components are neither on nor off, instead they are kept in-between these two states, i.e., components are kept in a reduced readiness state until required.

The Functional dependency (FDEP) gate represents the functional dependency of some events to another trigger event. This gate helps to design a scenario when the operations of some components of a system are dependent on the operation of another component of the system. For example, when many components of a system receive power from a single source of supply, then failure of the power supply would cause all the dependent components to fail. In the FDEP gate there is only one trigger event (either a basic event or an intermediate event) but there could be multiple functionally dependent

events. As illustrated, the event T is the trigger event and the events A, B, and C are the dependent events, and they will fail if T occurs. In other words, those events (A, B, and C) are functionally dependent to event T. However, they can have their own individual failure, which will not affect the occurrence of the trigger events. The FDEP gate is particularly useful for modelling networked systems, where communication between connected components takes place through a common network element, and failure of the common element isolates other connected components. This type of gate can also model interdependencies, which would otherwise introduce loops in the fault trees.

2.3 Dynamic Fault Tree Analysis Methodologies

As DFTs introduce dynamic gates in classical fault trees, the typical combinatorial analysis techniques available for classical fault tree analysis cannot be directly applied to analyse DFTs. Several methodologies have been developed for both qualitative and quantitative analyses of DFTs. Qualitative analysis mainly focuses on determining cut sequences from DFTs.

On the other hand, quantitative analysis aims at determining the probability of the top event given the failure rate or failure probability or failure probability distribution of the basic events of the DFTs. Additionally, criticality analysis of events is also performed as part of quantitative evaluation of DFTs. The approaches used for developing methodologies for DFT analysis include, but not limited to, Markov models, Petri Nets, Bayesian Networks, Analytical solution, and Monte Carlo simulation. In the following subsections, we briefly discussed the qualitative and quantitative analysis approaches for DFTs.

2.3.1 Qualitative Analysis of Dynamic Fault Trees

In a qualitative analysis of traditional static fault trees, minimal cut sets (MCSs) are determined from the fault tree structure. An MCS represents the minimal combination of events that can cause the top event of the fault tree. An MCS-based qualitative analysis of a DFT is possible if the dynamic gates of the DFT are replaced by static gates. For instance, by replacing the FDEP gates by OR gates and replacing PAND and SPARE gates by AND gates. However, in this case, the temporal dependencies between events would not be retained.

In (Xiang *et al.*, 2012), a method was proposed to allow combinatorial analysis of DFT with priority-AND gate only. In their work, the PAND gate was transformed to an

AND gate by adding some conditioning events and the new gate was called CAND. The work was later extended in (Xiang *et al.*, 2013).

To capture the temporal dependencies between events, the concept of minimal cut sequences (MCSQ) was proposed by (Tang and Dugan, 2004). An MCSQ is the minimal sequence of events that is sufficient and necessary to cause the top event of the DFT. To generate the cut sequences for a DFT, the zero-suppressed binary decision diagrams (ZSBDD) (Minato, 2001) were used. It was shown that the dynamic gates can be replaced by the static gates to determine the cut sets and then cut sequences can obtained by adding necessary sequencing information into the cut sets. Later, for cut sequence generation, (Liu *et al.*, 2007) proposed an algorithm called Cut sequence set algorithm (CSSA) using the notion of sequential failure symbol (SFS).

SFS is a mechanism to describe the sequential failure between two independent events. Later, the concept of the extended cut sequence was proposed based on the general cut sequence by (Zhang *et al.*, 2011). In the above approaches, the concept of cut sequence was under the assumption of non-repairability of system components. In (Chaux *et al.*, 2013), a new definition of cut sequences was provided for binary systems, i.e., the system can either be in working or in failed states, with repairable components.

In (Walker, 2009), Walker proposed a qualitative analysis approach for the Pandora temporal fault tree. He also provided temporal laws for to facilitate the minimization of the temporal sequences of events. One year later, (Merle, 2010) introduced an algebraic method for determining and expressing cut sequences of dynamic fault trees. This approach was based on the extension of the structure function used for classical static fault tree analysis. In (Rauzy, 2011), Rauzy introduced a variant of ZSBDD approach proposed in (Minato, 2001) to include sequencing information. This variant can be used for the determination of cut sequences of DFT.

In (Kabir *et al.*, 2017), a model-based approach was proposed for qualitative analysis of dynamic failure behaviour of systems. (Elderhalli *et al.*, 2017) integrated theorem proving and model checking to propose a comprehensive approach for qualitative and quantitative analysis of DFTs. Most recently, (Piriou, Faure and Lesage, 2019) provided a new definition of MCSQ for dynamic, repairable and reconfigurable systems. Afterwards, an algorithm was proposed to derive the MCSQs from Generalized Boolean logic Driven Markov Processes (GBDMP) (Piriou, Faure and Lesage, 2017) models.

2.3.2 Quantitative Analysis of Dynamic Fault Trees

A brief taxonomy of DFTs' quantitative solution techniques reviewed in this chapter is shown in Figure 2-2. The meaning of each sign has been explained at the bottom of the figure. As an example, in this figure, `R' sign stands for the ability to model and solve the repairable DFTs, `t' refers to a time-consuming procedure, and `D' means the solution is applicable for on-demand safety analysis.

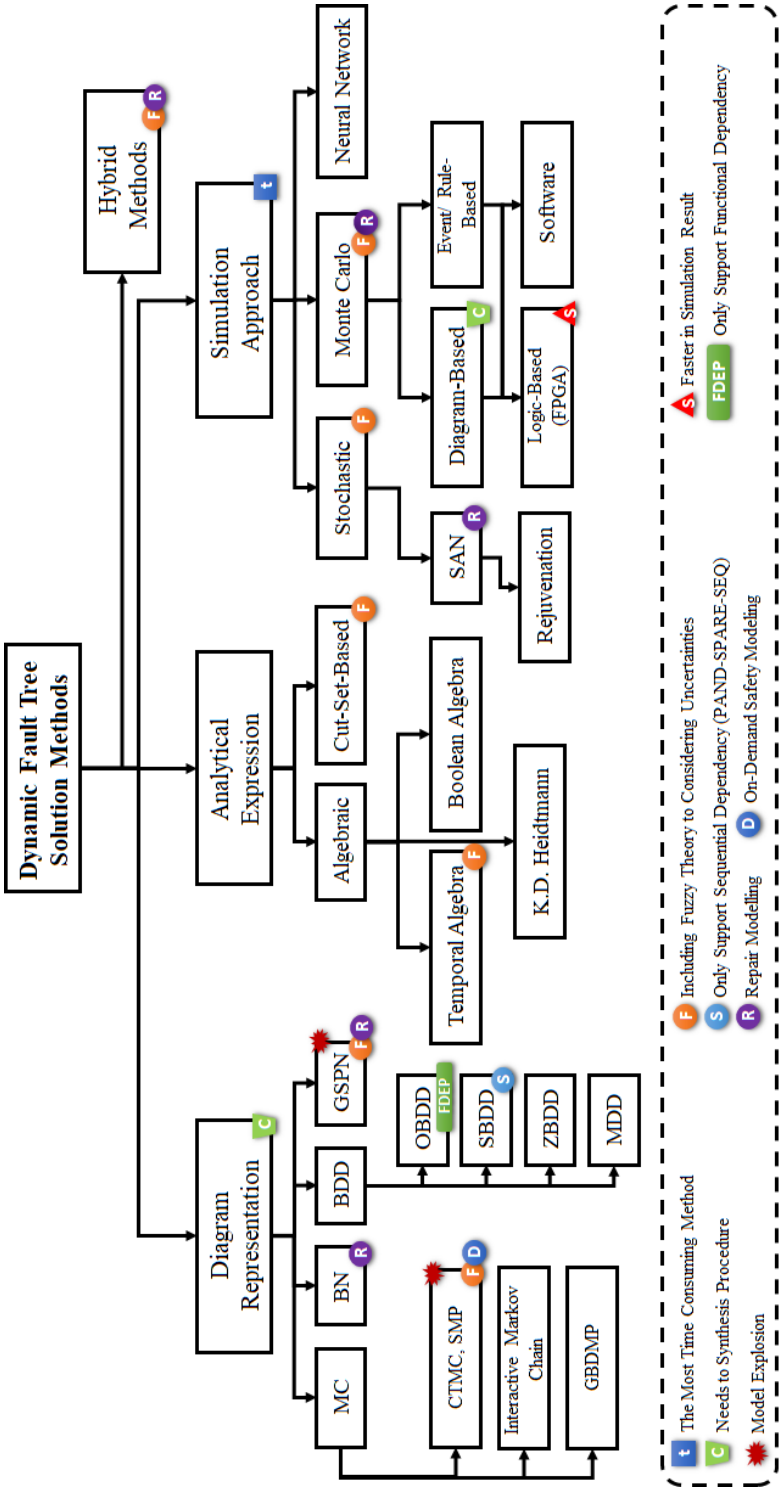


Figure 2-2. Taxonomy of existing solutions for DFTs (Aslansefat, Kabir, *et al.*, 2020)

2.3.3 Markov Models for quantifying DFTs

Solving the DFT by the use of Continuous-Time Markov Chain (CTMC) is regarded as one of the first and most important solution methods developed for quantitative evaluation of DFTs. This method has been employed in the structure of software tools such as Galileo, DIFtree and HiRel (Bavuso *et al.*, 1994; Dugan, Venkataraman and Gulati, 1997).

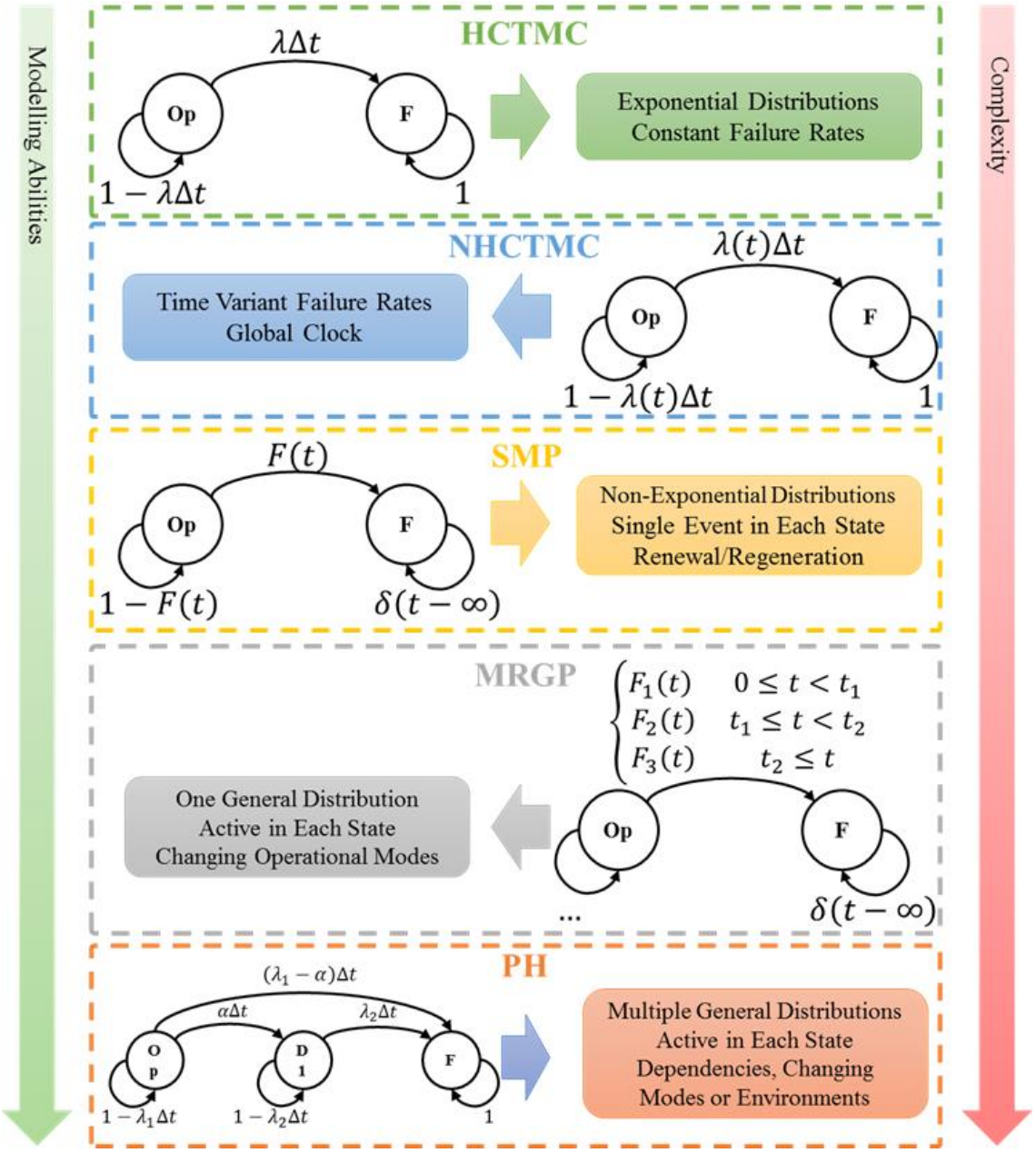


Figure 2-3. Classification of Markov Models (Aslansefat, Kabir, *et al.*, 2020)

As shown in Figure 2-3, Markov models can be categorized into five types; I) Homogenous Continuous Time Markov Chain (HCTMC) known as a traditional CMTC and it can model failures with exponential probability distribution with constant failure rates. II) The second type is the Non-Homogenous Continuous Time Markov Chain (NHCTMC) that can model global clock and exponential type failures with time variant failure rates. III) Semi-Markov Process (SMP) is the third category that enables to consider non-exponential probability distributions and renewal processes. IV) The fourth category is Markov Regenerative Process (MRGP) which is capable of considering operational mode changes in one transition. V) The Phased Type Markov Process (PH) is the last category and it can model multiple general distributions through diving the systems' states into some degraded states (more degraded states more accuracy) (Trivedi and Bobbio, 2017). It should be noted that there are some other extensions of Markov models such as Input/output Interactive Markov Chains and Generalized Boolean logic Driven Markov Processes (GBDMP) which are obtained from the combination of Markov theorem and Automata. In fact, each of those introduced Markov types can be merged with Automata or similar theories to generate the extended versions. In Figure 2-3, the modelling capability is increasing from top to bottom while the complexity of computation is also raising.

Having categorized Markov models, the use of these models for reliability evaluation of DFTs is briefly studied as follows. In 1991, the first concept of dynamic fault tree and its dynamic gates such as PAND, SPARE, SEQ and FDEP have been introduced through their CTMCs (Boyd, 1992). The reference also recommended an automatic way for conversion of DFT to its equivalent Markov Chain. Following this, in 1993, evaluation of the system behaviours considering imperfect coverage has been studied (Dugan, Bavuso and Boyd, 1993). Two benchmarks named Fault Tolerant Parallel Processors (FTPP) and Mission Avoidance Systems (MAS) that are used later by many researchers, were also introduced in this article. The reliability analysis of DFT in the presence of transient and permanent faults, failure dependencies, recovery of a system and reconfiguration of FTPP benchmark was studied in (Dugan, 1993). From 1993 to 2009 several studies have been performed to address different issues such as the accuracy of conversion procedure from DFT to CTMC (Manian *et al.*, 1999), uncertainty analysis (Yin, Smith and Trivedi, 2001), imperfect coverage consideration (Vesely *et al.*, 2002), decomposing DFTs into independent modules (Huang and Chang, 2007), introducing new Markov models for components' failures (Dominguez-Garcia *et al.*, 2008),

considering repeated events and their effects in state-space modelling (Yuge and Yanagi, 2008) in DFT-based reliability analysis.

In 2009, (Norberg *et al.*, 2009) presented a model for merging static fault tree with availability CTMC, so that it could evaluate the risk parameter. By the use of this method, reliability, risk, availability, failure rate, failure interval, MTBF and MTTF were induced from fault tree. This thesis employed this method on drinking water supply system. (Verma, Srividya and Karanki, 2010) studied different methods for reliability modelling and then discussed the behaviour of dynamic gates along with CTMC. In addition, they described DFT solutions by the use of CTMC and Monte Carlo theories. Although, in general, the CTMC-based approaches are applicable only to exponentially distributed data, (Guo, Han and Liu, 2011) proposed an approach combining failure rates with Weibull distribution with CTMC. (Zixian *et al.*, 2011) reported a widespread use of reliability methods in evaluating the risk of surgery and with this purpose, they evaluated time independent risk and time dependent risk through merging CTMC and static fault tree. By calculating failure rate of medical facilities, they evaluated surgery frequency, rescue timeliness and risk of gastric-oesophageal surgery using fault tree. Then by using sensitivity analysis, the effect of retrieval time factor and rescue timeliness was measured. A Power Factor Correction (PFC) using CTMC in DFT of power systems has been presented in (Ranjbar, Kiani and Fahimi, 2011).

In 2012, the Fuzzy-CTMC models have been proposed by (Li *et al.*, 2012) to solve the Fuzzy DFTs and evaluate their reliability under the condition of uncertainty. They presented an example of automatic hydraulic system cutting machine (CNC). Their study only considered a dynamic fault tree example with FDEP gate and fuzzy evaluation of other gates are left vague. This fuzzy approach was also used in another thesis for the reliability evaluation of driver in array of solar cells (Huang *et al.*, 2013). A year later, the statistical reliability evaluation of a dynamic fault tree with PAND gate has been proposed by (Xiang *et al.*, 2013) in which the conversion of the PAND gate into AND gate along with considering some dependent conditional events was introduced. Moreover, the newly introduced AND gate called CAND was assumed to be dependent upon conditional events. In this study, CTMCs for PAND and CAND gates were provided with a discussion about their differences and used in the reliability evaluation of FTPP's benchmark. The combination of BDD and CTMC for reliability evaluation of DFTs has been introduced in (Hao, Zhang and Wei, 2014).

The use of Shannon's decomposition theory has been proposed by (Ge and Yang, 2015) to solve DFTs. The proposed method increased the computational efficiency. However, the thesis only considered PAND gate, and the method was not generalized for other dynamic gates. (Brameret, Rauzy and Roussel, 2015) proposed a framework called "AltaRica" to reduce the state explosion through combining the Dijkstra's algorithm and notion of the distance factor for the DFT solution. An approximate solution for DFT through truncating Markov chain states has been presented in 2016 by (Yevkin, 2016). The method was appropriate for both repairable and non-repairable systems. In 2017, the research work of (Ge and Yang, 2015) has been extended and published in (Ge and Yang, 2017). The research has covered spare and sequence gates through De Morgan theorem, and for negating a generalized cut sequence, they have improved explicit formula. In 2018, a new state-space generation approach for solving the DFTs has been proposed by (Volk, Junges and Katoen, 2018). The presented method has the ability of model reduction through model checking theories.

A hierarchical and approximate solution for availability analysis in DFTs based on equivalent two-state Markov models has been proposed by (Ramezani *et al.*, 2016). Their approach was only tailored for exponential failure distribution-based events.

An automated tool for the evaluation of repairable DFT has been presented by (Manno *et al.*, 2014). The thesis proposed a mapping from DFT entity to adaptive transition system entity, and a conception of failure gates for the evaluation of both reliability and availability has been illustrated. This thesis used the SMP for reliability evaluation of DFTs. A novel hierarchical SMP-based solution for reliability assessment of DFTs was also proposed by (Aslansefat, 2014) in which the computational complexity and the state explosion of the SMP have decreased significantly.

As mentioned before, Input/Output Interactive Markov Chain (I/O IMC) is an extension for CTMC which is used for DFT solutions (refer to (Hermanns, 2002; Crouzen, 2006; Boudali, Crouzen and Stoelinga, 2007, 2010; Arnold, Belinfante, der Berg, *et al.*, 2013; Arnold, Belinfante, Van Der Berg, *et al.*, 2013). The use of I/O IMCs can reduce state space explosion. In addition, these models enable us to consider the standby spare behaviours in the basic events. Generalized Boolean logic Driven Markov Processes (GBDMP) another extension of Markov Process has been also used for qualitative and quantitative analysis of the DFT by (Piriou, Faure and Lesage, 2017). Moreover, Sequential Binary Decision Diagram (SBDD) and its extensions have been

used in (Tannous, Xing and Dugan, 2011; Xing, Shrestha and Dai, 2011; Xing, Tannous and Dugan, 2012; Ge *et al.*, 2015, 2016) for quantitative evaluation of DFTs.

Markov process has been used by (Niwas and Garg, 2018) to propose an approach to evaluate the reliability, availability of an industrial system under the cost-free warranty policy, where the working period of a system is followed by a rest period. To address the issue of uncertain failure data in Markov chain-based reliability evaluation, (Garg, 2015) used a fuzzy Markov model of a repairable system to develop the n^{th} order fuzzy Kolmogorov's differential equations. Later the fuzzy reliability of the system both in transient and steady state was evaluated using Runge--Kutta method. (Aslansefat and Latif-Shabgahi, 2020) proposed a novel hierarchical SMP-based approach as a solution for reliability evaluation of DFTs.

In (Kabir *et al.*, 2020) proposed a hybrid approach in which Markov-based solutions along with algebraic and Petri-nets where used in a hybrid way to increase the efficiency of the solution. A Fuzzy Markov-based solution has been introduced by (Yan *et al.*, 2021) for applied for reliability evaluation of an autonomous computer system. The method was limited in its applicability especially dealing with repeated basic events. (Andrews and Tolo, 2023) has proposed a hybrid approach utilizing Markov process, BDD and Petri Nets to evaluate the reliability of the Dynamic Fault Tree considering repeated BEs and non-exponential failures. The work has four main phases of Contraction 1, Factorization, Extraction, Contraction 2.

The thesis presented a number of hypothetical and industrial examples. It also has an example related to the repair consideration in DFTs and its SMP-based solution. Considering the addressed state-of-the-art for Markov-based quantitative DFT solution, it can be highlighted that the proposed approach in this thesis has its own unique characteristics such as considering Complex BEs, including runtime evaluation and using Markov path to reduce computation complexity.

2.4 Semi-Markov Transient Analysis for Reliability Evaluation

In this section, analytical solution of Semi-Markov Process (SMP) is addressed and then the reliability of PAND gate is calculated by this theorem. SMP can be modelled by different notations (Zajac & Kierzkowski, Attempts at Calculating Chosen Contributors with Regard to the Semi-Markov Process and the Weibull Function Distribution, 2011), This project uses tuple (trio) (p, P, F(t)), where: p is vector of initial distribution, P is

matrix of conditional transition probabilities and F(t) describes matrix of distribution functions of sojourn times in state i th, when j th state is next;

Considering X_i, $\forall i = 0,1,2,...$ as a random variable, the time-homogeneous SMP X is determined by a vector of initial state probabilities $p(0) = [P\{X_0 = i\}] = [1,0,...,0]$, and the conditional transition probability matrix $P(t) = [P_{ij}(t)]$. The conditional probabilities' matrix ($P_{ij}(t)$ is satisfied by Kolmogorov-Feller's equations in (2-1) (Kulkarni, 1995).

$$P_{ij}(t) = \delta_{ij}[1 - G_i(t)] + \sum_{K \in S} \int_0^t P_{kj}(t-x) dQ_{ik}(x) \quad (2\text{-}1)$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise, G_i is the distribution of the sojourn time in state i (Yin, Fricks, & Trivedi, 2002; Frattini, Bovenzi, Alonso, & Trivedi, 2013), and $Q_{ij}(t)$ describes the kernel matrix. Note that the kernel matrix has a Markov renewal theorem in the background (Fricks, Telek, Puliafito, & Trivedi, 1998).

Solution of (2-1) an be found by applying Laplace Stieltjes Transformation (LST) in (2-3) (Rios Insua, Ruggeri, & Wiper, 2012). This is a set of Voltera equations in (2-2) which is a Markov renewal equation (Yin, Fricks, & Trivedi, 2002). Note that for non-exponential failure distributions such as Weibull and Gamma, some approximation is needed (Refer to (Zajac & Budny, On Determination of Some Characteristics of Semi-Markov Process for Different Distributions of Transient Probabilities, 2009; Zajac & Kierzkowski, Attempts at Calculating Chosen Contributors with Regard to the Semi-Markov Process and the Weibull Function Distribution, 2011; Distefano, Longo, & Trivedi, 2012)).

$$\tilde{p}_{ij}(s) = \delta_{ij}[1 - \tilde{g}_i(s)] + \sum_{K \in s} \tilde{q}_{ik}(s)\tilde{p}_{kj}(s) \quad (2\text{-}2)$$

Equation (2-2) in the matrix form can be rewritten as follows:

$$\tilde{p}(s) = [I - \tilde{g}(s)] + \tilde{q}(s)\tilde{p}(s) \quad (2\text{-}3)$$

Hence, it can be rewritten as (2-4) through simple algebraic replacement.

$$\tilde{p}(s) = [1 - \tilde{q}(s)]^{-1}(I - \tilde{g}(s)) \quad (2\text{-}4)$$

In (2-4), the inverse of $1 - \tilde{q}(s)$ can be replaced by the summation of powers of $\tilde{q}(s)$. The resulted equation which is useful for a singular kernel matrix will be as (2-5).

$$\tilde{p}(s) = \left(\sum_{n=0}^{\infty} \tilde{q}(s)^n\right)(I - \tilde{g}(s)) \quad (2\text{-}5)$$

Having solved (2-5) with taking the inverse LST of $\tilde{p}(s)$, the unconditional state probabilities in time domain are determined as follows:

$$P(t) = P(0)P(t) \tag{2-6}$$

Finally, the reliability of system can be achieved by summing of the transient probability of operational states.

2.5 Reliability Evaluation of PAND Gate through SMP

The Markov model of PAND gate has previously been presented in the literature. Figure 2-4 illustrates the semi-Markov model of PAND gate with any failure distribution function. It can be solved by SMP theorem. In this model, $F_A(t)$ is Cumulative Distribution Function (CDF) of the first input of PAND gate and $F_B(t)$ is CDF of the second input of PAND gate respectively. States are numbered from 1 to 5 from top to bottom and left to right. This model can be solved by SMP theorem described in the previous sub-section.

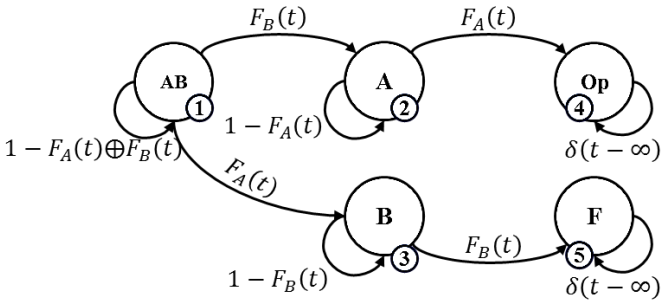


Figure 2-4. Semi-Markov model of a PNAD gate

The kernel matrix of Figure 2-4 can be written in the form of (2-7) and distribution matrix of the sojourn time in each state is in the form of (2-8).

$$Q(t) = \begin{bmatrix} 0 & Q_{1,2} & Q_{1,3} & 0 & 0 \\ 0 & 0 & 0 & Q_{2,4} & 0 \\ 0 & 0 & 0 & 0 & Q_{3,5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2-7}$$

The dimension of both kernel matrix and sojourn distribution matrix should be n by n where n is the number of states.

$$G(t) = diag\left(F_A(t) \oplus F_B(t), F_A(t), F_B(t), \delta(t-\infty), \delta(t-\infty)\right) \tag{2-8}$$

where the $\oplus$ operation is defined by (2-9). In other words this is a probabilistic OR algebraic operation.

$$A \oplus B = 1-(1-A)(1-B) \tag{2-9}$$

In matrix (2-7), $Q_{1,2}$ is the system failure probability up to time t, if B is occurred at first. It can be written as (2-10). Subscript 1,2 stands for “from state one to state two”.

$$Q_{1,2} = \Pr\{X_1 = 2, S_1 \le t \mid X_0 = 1\}$$
$$= \Pr\{L_B \le t \wedge L_A \le L_B\} = \int_0^t 1 - F_A(t)\, dF_B(t) \quad (2\text{-}10)$$

For exponential failure behaviour case of A and B (2-10) can be written as (2-11) where λ_A and λ_B are the constant failure rates of events or inputs A and B respectively.

$$Q_{1,2} = \int_0^t \left[1 - \left(1 - e^{-\lambda_A \tau}\right)\right] d\left\{1 - e^{-\lambda_B \tau}\right\}$$
$$= \int_0^t e^{-(\lambda_A + \lambda_B)\tau} \lambda_B \, d\tau = \frac{\lambda_B \left[1 - e^{-(\lambda_A + \lambda_B)t}\right]}{\lambda_A + \lambda_B} \quad (2\text{-}11)$$

Similarly, $Q_{1,3}$ is the system failure probability up to time t, in a case that A is occurred first. It means the related occurrence time of event A is less than related occurrence time of event B. It can be written as (2-12). For exponential failure distribution function of both inputs, (2-12) can be written as (2-13).

$$Q_{1,3} = \Pr\{X_1 = 3, S_1 \le t \mid X_0 = 1\}$$
$$= \Pr\{L_A \le t \wedge L_B \le L_A\} \quad (2\text{-}12)$$

Similarly, λ_A and λ_B are the constant failure rates of events or inputs A and B respectively.

$$Q_{1,3} = \int_0^t \left[1 - \left(1 - e^{-\lambda_B \tau}\right)\right] d\left\{1 - e^{-\lambda_A \tau}\right\}$$
$$= \int_0^t e^{-(\lambda_A + \lambda_B)\tau} \lambda_A \, d\tau = \frac{\lambda_A \left[1 - e^{-(\lambda_A + \lambda_B)t}\right]}{\lambda_A + \lambda_B} \quad (2\text{-}13)$$

$Q_{2,4}$ is the failure probability of input or event A before or at time t and $Q_{3,5}$ is the failure probability of input or event B before or at time t. $Q_{2,4}$ and $Q_{3,5}$ are defined by (2-14) and (2-15), respectively.

$$Q_{2,4} = \Pr\{X_1 = 4, S_1 \le t \mid X_0 = 2\}$$
$$= \Pr\{A \text{ fails before or at time t}\}$$
$$= \Pr\{L_A \le t\} \quad (2\text{-}14)$$

Note that 2,4 denotes “from state two to state 4” and 3,5 denotes “from state three to state five”.

$$Q_{3,5} = \Pr\{X_1 = 3, S_1 \le t \mid X_0 = 5\}$$
$$= \Pr\{B \text{ fails before or at time t}\}$$
$$= \Pr\{L_B \le t\} \quad (2\text{-}15)$$

In exponential form (2-14) and (2-15) are written by (2-16) and (2-17), respectively.

$$Q_{2,4} = = \int_0^t \lambda_A e^{-\lambda_A \tau} d\tau = 1 - e^{-\lambda_A t} \quad (2\text{-}16)$$

It should be noted that both failure rates are constant.

$$Q_{3,5} == \int_0^t \lambda_B e^{-\lambda_B \tau} d\tau = 1 - e^{-\lambda_B t} \tag{2-17}$$

The LST of kernel matrix is written as (2-18) in its general form. As can be seen, in this matrix, there are four nonzero transitions.

$$\tilde{q}(s) = \begin{bmatrix} 0 & L\{Q_{1,2}\} & L\{Q_{1,3}\} & 0 & 0 \\ 0 & 0 & 0 & f_A^*(s) & 0 \\ 0 & 0 & 0 & 0 & f_B^*(s) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2-18}$$

In general form, the LST of G matrix is written as (2-19). In exponential form, it is written as (2-18). Note that "*diag*" creates diagonal matrix from each input vector.

$$\tilde{g}(s) = diag\left(L\{F_A(t) \oplus F_B(t)\}, f_A^*(s), f_B^*(s), 1, 1\right) \tag{2-19}$$

It is assumed that the failure rates are constant in the following equation.

$$I - \tilde{g}(s) = diag\left(\frac{s}{s+\lambda_A+\lambda_B}, \frac{s}{s+\lambda_A}, \frac{s}{s+\lambda_B}, 1, 1\right) \tag{2-20}$$

Similarly, (2-21) is LST of kernel matrix for exponential failures.

$$\tilde{q}(s) = \begin{bmatrix} 0 & \frac{\lambda_B}{s+\lambda_A+\lambda_B} & \frac{\lambda_A}{s+\lambda_A+\lambda_B} & 0 & 0 \\ 0 & 0 & 0 & \frac{\lambda_A}{s+\lambda_A} & 0 \\ 0 & 0 & 0 & 0 & \frac{\lambda_B}{s+\lambda_B} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2-21}$$

By the use of (2-4) or (2-5), the unconditional probability vector of Figure 2-4 is computed as (2-22).

$$\begin{aligned} \hat{P} = \Big[& L^{-1}\left\{\left(1 - f_A^*(s)\right)\left(1 - f_B^*(s)\right)\right\}, \\ & L^{-1}\left\{q_{1,2}(s)\left(1 - f_A^*(s)\right)\right\}, , L^{-1}\left\{q_{1,3}(s)\left(1 - f_B^*(s)\right)\right\} \\ & L^{-1}\left\{q_{1,2}(s)\left(f_A^*(s)\right)\right\}, \quad L^{-1}\left\{q_{1,3}(s)\left(f_B^*(s)\right)\right\} \Big] \end{aligned} \tag{2-22}$$

Finally, reliability of PAND gate can be obtained from the probability of fail state (state 5 in Figure 2-4) using (2-23). We named this equation as "general equation of PAND gate", Note that, this project defines a general equation for each dynamic gate of a given DFT obtained by SMP.

$$R(t) = 1 - F(t) = 1 - L^{-1}\left\{q_{1,3}(s)\left(f_B^*(s)\right)\right\} \tag{2-23}$$

The final reliability expression shows the consistency of the results with the existing one in the literature.

$$R(t) = 1 - F(t) = 1 - L^{-1}\left\{\left(\frac{\lambda_A}{s+\lambda_A+\lambda_B}\right)\left(\frac{\lambda_B}{s+\lambda_B}\right)\right\}$$
$$\Rightarrow R(t) = e^{-\lambda_B t} + \frac{\lambda_B\left(1-e^{-(\lambda_A+\lambda_B)t}\right)}{\lambda_A+\lambda_B} \quad (2\text{-}24)$$

2.6 Markov Steady-state Analysis for Availability Evaluation

Markov Regenerative Processes are founded on Markov renewal theory and enable complex state transitions in a state space with the general distribution. The MRGP can be considered as a state-space stochastic process with the ability to restart itself in a probabilistic domain. In this process, it is not necessary to have the Markov property between regeneration epochs while the Markov property will be satisfied in the sequence of regeneration time points (Machida, Xia and Trivedi, 2018). Note that at the regeneration point, the process state does not depend on the history of the previous point. For more information regarding the renewal processes and MRGP, see (Kulkarni, 2016).

Consider the stochastic process $X_{n\text{with n}\backslash geq0}$ that follows the MRGP properties. It has also a number of states that are defined in the state space Ω. The regeneration epochs can be formed by T_n where $n \geq 0$. The kernel distribution representing the former process can be defined as (2-25) that considers the conditional probabilities going from state i to state j in state-space Ω.

$$K_{\{ij\}(t)} = \Pr\{X_{\{n+1\}} = j, T_{\{n+1\}} - T_n \leq t \mid X_n = t\} \quad (2\text{-}25)$$

Given kernel distributions for all transitions in a defined MRGP, the global kernel matrix of the MRGP can be formed. Similarly, the local kernel distribution can be formed as (2-26) and captures the behaviour or distribution between two consecutive regeneration time points when the process is in state $j \in \Phi$ at time t starting from state $i \in \Omega$ (Machida, Xia and Trivedi, 2018). It can also be defined as a matrix considering all possible local distributions.

$$E_{ij}(t) = Pr\{Z(t) = j, T_1 > t \mid Z(0) = i\} \quad (2\text{-}26)$$

The mean sojourn time in state j considering the state i as initial regeneration state can be calculated by (2-27).

$$\alpha_{ij} = \int_0^\infty E_{ij}(t)dt \quad (2\text{-}27)$$

Considering the finite and irreducible model, the steady-state probability vector v is given by solving the following linear system. Note that the summation of all elements in this vector should be equal to one.

$$\mathrm{v} = \mathrm{v} \cdot K(\infty) \tag{2-28}$$

The steady-state probability of state j can be obtained from (2-29).

$$\pi_j = \frac{\sum_{k\epsilon\Omega} v_k \alpha_{kj}}{\sum_{k\epsilon\Omega} v_k \sum_{l\epsilon\Omega} \alpha_{kl}} \tag{2-29}$$

For demonstration, a simple example with three-state MRGP follows. Consider the RGMP of Figure 3-5 with hypothetical parameters and distributions, where two failure transitions are Weibull and defined as $F(t) = 1 - e^{-(t/\beta_1)^{\alpha_1}}$ and $G(t) = 1 - e^{-(t/\beta_2)^{\alpha_2}}$. Repair transitions consider a deterministic function defined as (2-30). Assume that $\alpha_1 = 2$, $\alpha_2 = 0.75$, $\beta_1 = 2000$, $\beta_2 = 10000$, $T_1 = 1$ and $T_2 = 2$ (Distefano, Longo and Trivedi, 2012a).

$$R(t) = \begin{cases} 0 & t < T_1 \\ 1 & t \geq T_1 \end{cases} \quad and \quad \acute{R}(t) = \begin{cases} 0 & t < T_2 \\ 1 & t \geq T_2 \end{cases} \tag{2-30}$$

To simplify the problem, consider the “Op”, “PM” and “F” states as “1”, “2” and “3” respectively. The kernel distribution from state “1” to state “2” can be obtained using (2-25) as (2-31).

$$K_{12}(t) = \int_0^t \big(1 - F(t)\big)\, dG(t) \tag{2-31}$$

Similarly, the kernel distribution from state “1” to state “3” can be achieved as (2-32).

$$K_{13}(t) = \int_0^t \big(1 - G(t)\big)\, dF(t) \tag{2-32}$$

Two other kernel distributions will be $K_{21}(t) = \acute{R}(t)$ and $K_{31}(t) = R(t)$. Having obtained all kernel distributions, for the embedded Markov chain, the probability transition matrix can be formed as:

$$P = K(\infty) = \begin{bmatrix} 0 & K_{12}(\infty) & K_{13}(\infty) \\ K_{21}(\infty) & 0 & 0 \\ K_{31}(\infty) & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \psi & 1-\psi \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{2-33}$$

where $\psi = \frac{\alpha_1}{\beta_1} \int_0^\infty e^{-(t/\beta_1)^{\alpha_1}} \left(\frac{t}{\beta_1}\right)^{\alpha_1 - 1} e^{-(t/\beta_2)^{\alpha_2}} dt$. Now, the sojourn time distributions should be obtained using (2-26).

$$E_1(t) = \big(1 - G(t)\big)\big(1 - F(t)\big), E_2(t) = 1 - \acute{R}(t) \; and \; E_3 = 1 - R(t) \tag{2-34}$$

Using (2-28), the mean sojourn time at each state can be calculated.

$$\alpha_1 = \int_0^\infty e^{-(t/\beta_1)^{\alpha_1} - (t/\beta_2)^{\alpha_2}} dt, \; \alpha_2 = T_1, \alpha_3 = T_2 \tag{2-35}$$

Considering $v_1 + v_2 + v_3 = 1$, and solving the linear system mentioned in (2-28):

$$v_1 = 0.5, v_2 = \psi/2, v_3 = (1 - \psi)/2 \tag{2-36}$$

Finally, the steady-state availability of the system can be achieved using (2-29) using $\alpha_1 = 2$, $\alpha_2 = 0.75$, $\beta_1 = 2000$, $\beta_2 = 10000$, $T_1 = 1$ and $T_2 = 2$.

$$A(\infty) = \pi_1 = \frac{v_1 \alpha_1}{\alpha_1 + \psi T_1 + (1 - \psi) T_2} = 0.999172 \tag{2-37}$$

2.7 Comparing with Monte Carlo Simulation

To validate the correctness and functionality of the HDFT solution, a Monte Carlo Simulation is used. In this simulation, the exponential failure and repair distribution are considered, and the inverse transform is applied to convert failure, repair, and delay rates to time-to-failure, time-to-repair and delay time. In addition, the RUL is executed directly in the simulation. As an example of the inverse transform, consider a component with an exponential failure distribution and the failure rate of λ. The time-to-failure can be obtained from (2-38) (Manno *et al.*, 2012).

$$t = -\frac{ln(1 - r)}{\lambda} \tag{2-38}$$

where r is a random number generated with uniform distribution. Similarly, the inverse distribution can be achieved for Weibull distribution as (2-39) (de Gusmão, Ortega and Cordeiro, 2011).

$$t = -\beta [ln(1 - r)]^{1/\alpha} \tag{2-39}$$

Regarding inverse transformation of the other failure distributions (Chiacchio *et al.*, 2013) can be referred. Simulating the aforementioned example, the mean value of the steady-state availability is achieved as 0.9989 with the variance of 0.0003. The consistency of the results in both simulation and the analytical approach can be used for model validation.

2.8 Computational Complexity Analysis

As SMPs and MRGP are using embedded Markov chain, their computation complexity can be described as O((x[00] + x[01])n) in which n represents the number of embedded Markov chain's states, x[00] stands for the number of non-zero transitions between states where the system is functional (the system is up) and x[01] stands for the number of transitions from operational/up states to failed/down states(Ciardo *et al.*, 1990). But in the proposed approach this high complexity is restricted to small component models. At system level, the proposed approach uses hierarchical FT evaluation, which means that the computation complexity can be simplified as O(k) where k is the number of gates in FT (Aslansefat and Latif-Shabgahi, 2020). In should be noted that the computational complexity of Continuous-Time Markov Chains (CTMC) without considering hierarchical evaluation can be described as $O(n^2)$ where n denotes the number of states (Ciardo *et al.*, 1990). In fact, MRGP is more complex than a CTMC but proposing the hierarchical method with the assumption of independence made can reduce its complexity for large-scale systems. It can be even less complex that CTMC for large-scale systems.

2.9 Proposed Markov Path Theorem

In this section, a novel methodology called Markov path theorem for reliability analysis of dynamic and complex systems is introduced through a number of examples.

2.9.1 Extended Markov Model (EMM)

The type of transition definition is the one of main differences of CTMC and SMP. In CTMC transitions are defined by constant rate or constant probability while in SMP transitions are defined by Cumulative Distribution Function (CDF). In this study, Extended Markov Model is introduced in which transitions are defined by Kernel Distribution Function (KDF). In order to explain the way of EMM modelling and its calculation, consider a hypothetical single-component system which could be modelled by three-states CTMC as Figure 2-5-(a).

In this model state A delineate operational situation of the system, state F depict failure of the system and state X is a hypothetical degraded operational situation of the system. Is assumed that system transits from state A to Failure state (F) with the rate of λ_1 and transits from state A to state X with rate of λ_2. Both mentioned rates are belonging to the exponential failure CDF. If the SMP theorem is considered, the SMP model of the

system can be illustrated as Figure 2-5-(b) in which transitions are obey from exponential CDF.

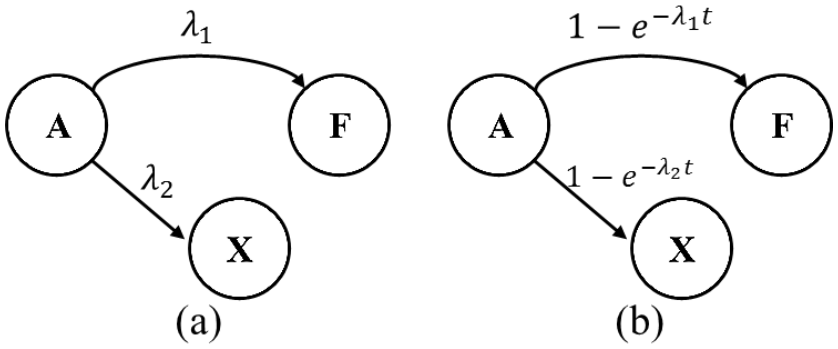


Figure 2-5. Hypothetical single-component system (a) CTMC model (b) SMP model

The above-mentioned models can be replaced by EMM as Figure 2-6. In the EMM model transitions are KDF and the way of their calculation will be addressed in the following. Note that each EMM is constructed from equivalent SMP model. In the following figure, the $Q_{1|\bar{2}}(t)$ can be called as "KDF transition".

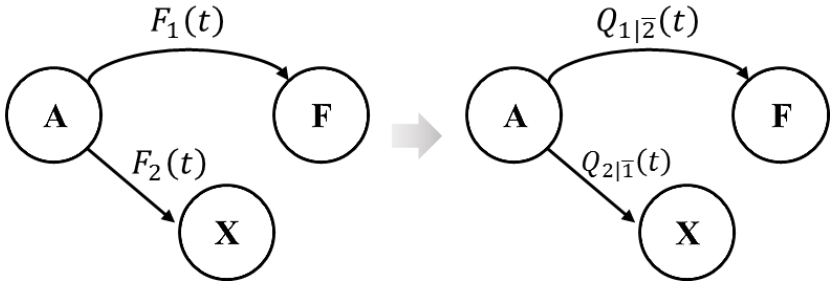


Figure 2-6. Extended Markov Model (EMM) of hypothetical single-component system

In case of exponential CDF, the EMM of the system can be built as Figure 2-7 in a similar way that the equation (2-11) obtained.

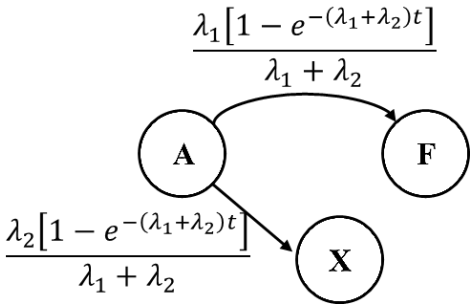


Figure 2-7. EMM of hypothetical single-component system when exponential CDFs are embedded.

It is suggested that before building of a EMM from a SMP model, if there are parallel transitions between two specific states of the SMP model, obtain the equivalent single transition by the use of the following equation.

$$F_{eq}(t) = 1 - \prod_{i=1}^{n} \left(1 - F_i(t)\right) \tag{2-40}$$

As an example, conversion of parallel transitions to single transition depicted as Figure 2-8 in which equivalent transition CDF can be obtained from the following equation.

$$\begin{aligned} F_{eq}(t) &= F_1(t) < OR > F_2(t) \\ &= 1 - \left(1 - F_1(t)\right)\left(1 - F_2(t)\right) \end{aligned} \tag{2-41}$$

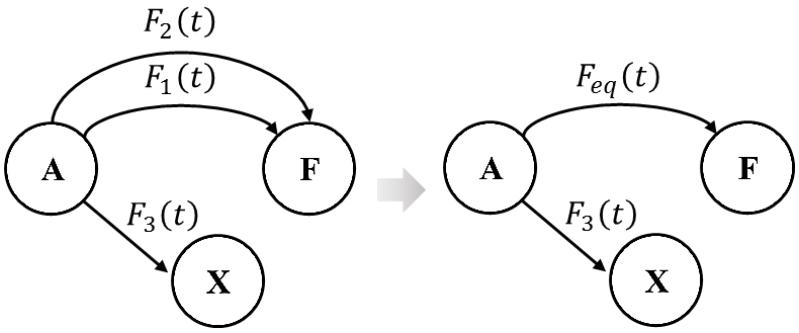


Figure 2-8. Conversion of parallel transitions to single transition before EMM building.

2.9.2 LST-based Extended Markov Model

Having built EMM, all kernel-based transitions can be replaced by Laplace Stieltjes Transform (LST) (see (Rios Insua, Ruggeri, & Wiper, 2012)) of that KDF. In the following, it will be shown that this conversion makes calculations easier and decrease problem complexity. Figure 2-9 conversion of EMM to LST-based EMM in general case.

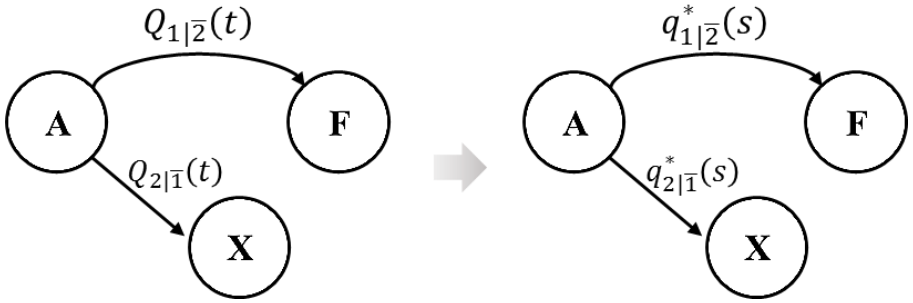


Figure 2-9. Conversion of EMM to LST-based EMM in general case

When transitions obey from exponential CDF, LST-based EMM can be obtained as Figure 2-10.

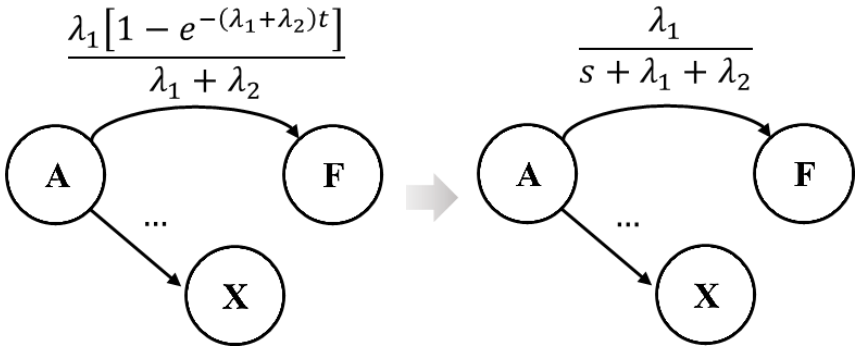


Figure 2-10. Conversion of EMM to LST-based EMM when transitions obey exponential CDF.

2.9.3 Definition of Markov Path

Consider a hypothetical Markov model and choose two different states of the model. Then mark one of them as a source state and mark another as a sink state. Markov Path defines as a set of non-repeated transitions which associate source state to the sink state. It is obvious that in each Markov model a number of paths between two different states may exist. Usually, in reliability applications, a state which describes a system with fully operational condition (all components of the system work perfectly) is chosen as the source state and failure state is chosen as the sink state.

The following figure illustrates the overall procedure for obtaining reliability using Markov path theorem. In this procedure the first step is to build the SMP or MRGP model of the system. The model then can be converted to its equivalent EMM and its transitions can be converted to LST-based EMM. Once the model is converted to LST-based EMM, it can be broken into several paths and to calculate the reliability, it is only needed to use the path that goes to the failure state of the system (the remaining parts of the model can be removed). At the end the reliability in time domain can be obtained using LST Inverse.

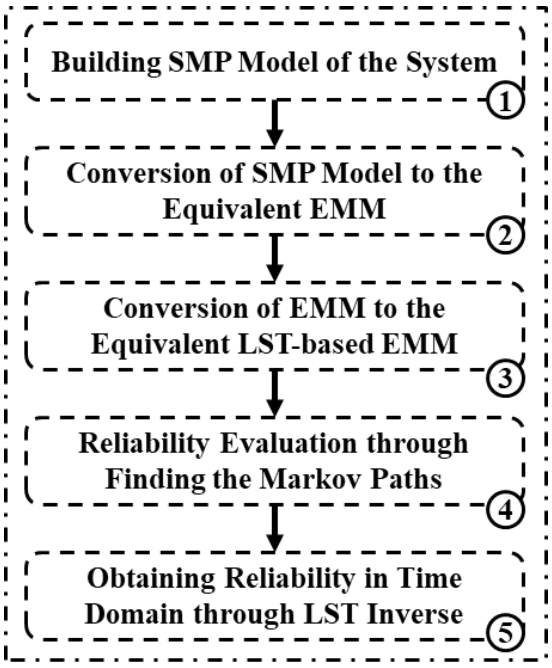


Figure 2-11. The overall procedure of obtaining reliability using Markov Path

To have an example, please consider a POR gate and its Markov model. The following figure illustrates how the POR gate model can be simplified using the Markov Path theorem.

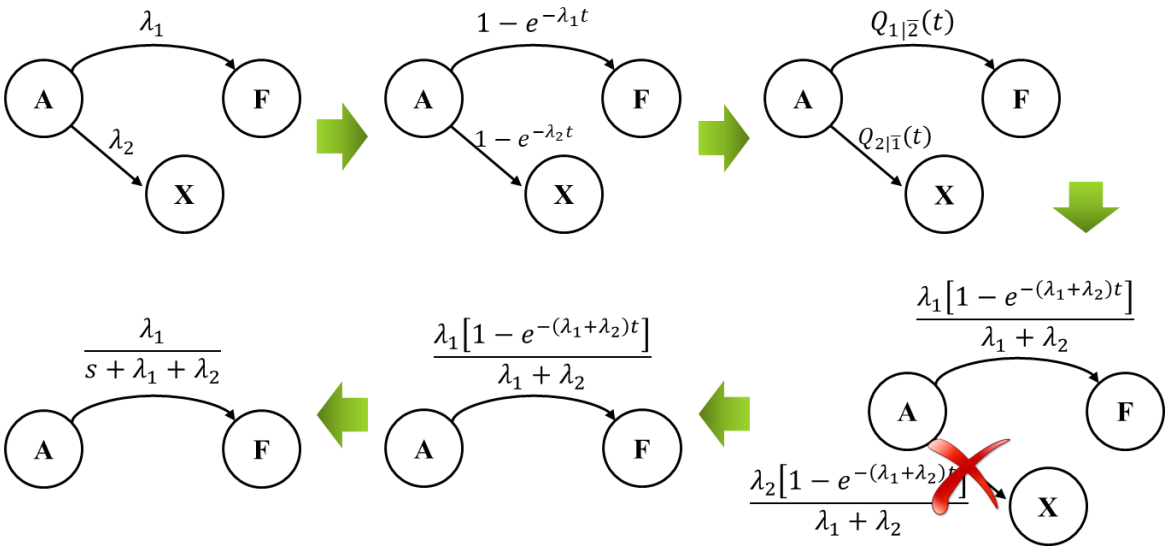


Figure 2-12. An example of POR gate and its simplification using Markov Path

The other example can be a cascaded PAND gate or a PAND gate with three inputs. As can be seen in the following model, a cascaded PAND gate with inputs A, B and C can be modeled by a Semi-Markov model with 12 states of which there are two absorbing states of failure and being operational. As shown in this figure, converting the model

using Markov path can simplify the model into a four states model which is easier to solve.

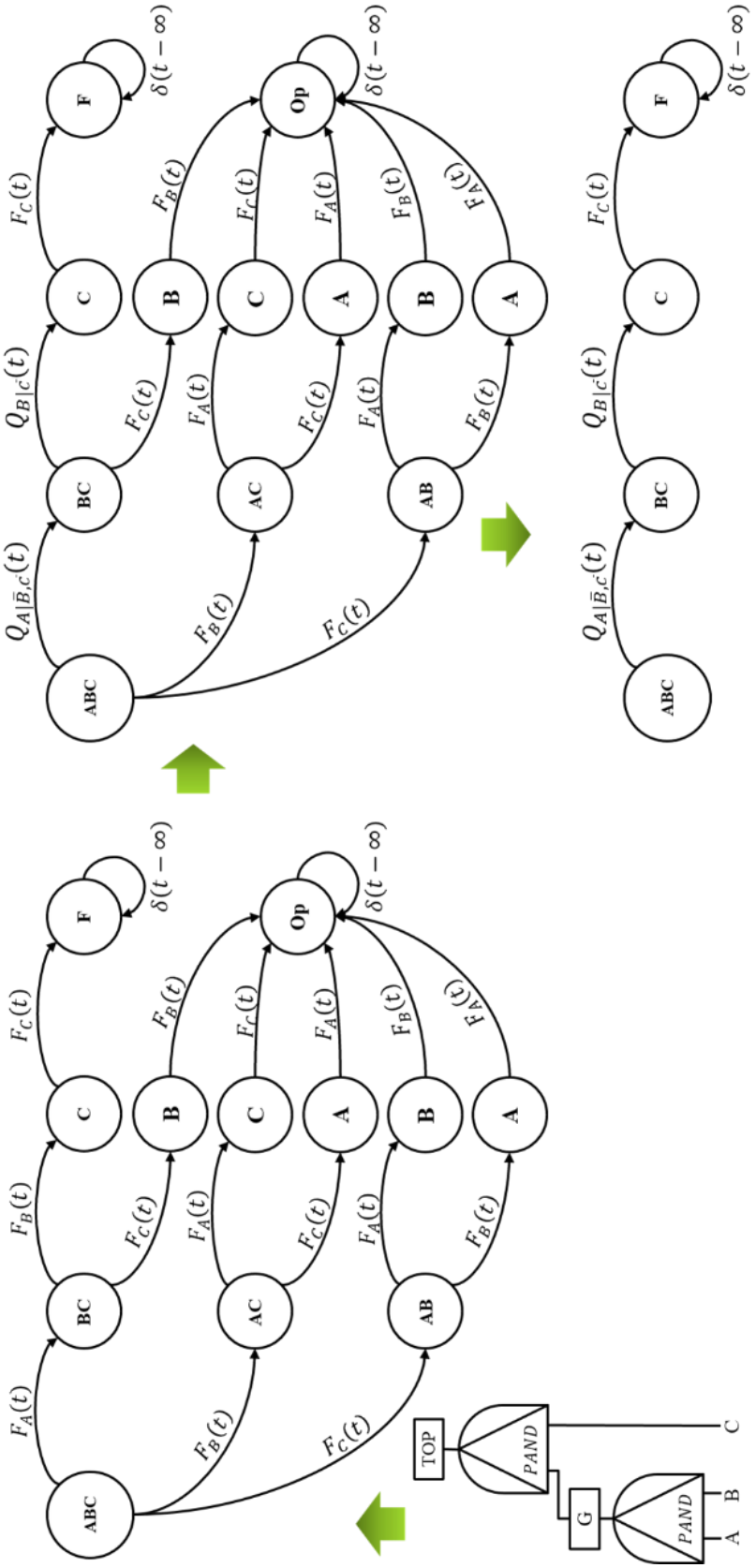


Figure 2-13. An example of a cascaded PAND gate and its simplification using Markov Path

2.10 Hierarchical Markov Models, a Solution for Dynamic Fault Tree

Dynamic Fault Trees (DFTs) was introduced, for the first time, by (Boyd, 1991) in which the whole DFT converted into a Markov model, and consequently, model explosion is resulted in. The problem was that the solution of this model was difficult and time-consuming. Later, a number of techniques based on Bayesian Networks, Petri Nets, Algebraic and etc. are presented for solving DFTs. In this chapter we present a hierarchical solving method for DFTs, based on SMP theorem.

First, the given DFT divided into a number of layers *(n)*. The lowest layer includes basic events and their associated gates. The outputs of the lowest layer form the inputs of first layer. The first layer contains a number of gates, the outputs of these gates form the input of second layer … the outputs of *(n-1)th* layer give the inputs of *(n)th* layer (Top layer). Starting from first layer, the gates of each layer are solved based on the method explained in previous sections to find their output CDF. The CDFs are used as $F(t)$ of inputs of the next layer. This procedure continuous until the CDF of Top event is obtained. It is now obvious that $R(t) = 1 - output\ CDF$. The procedure is depicted in the following figure.

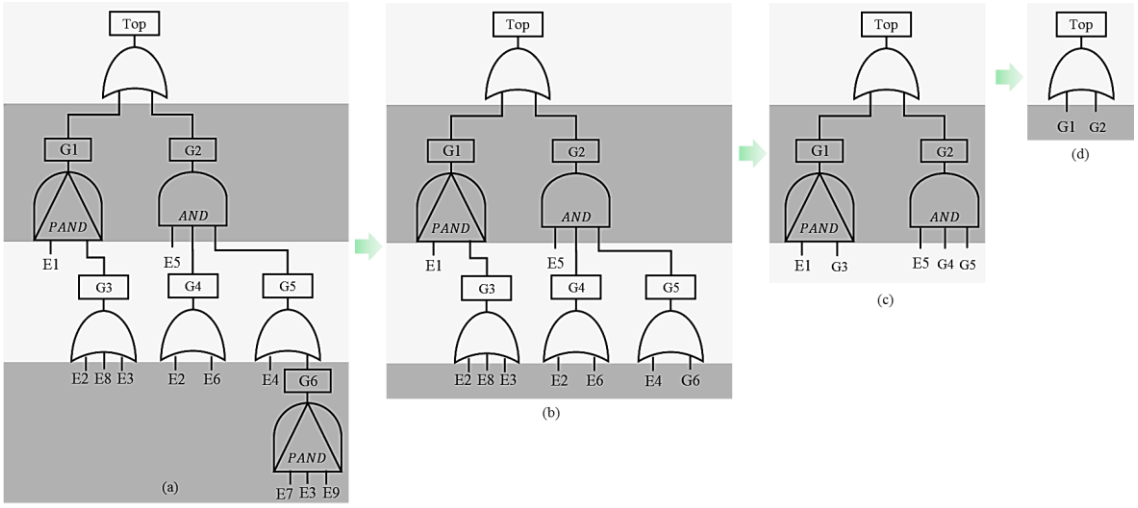


Figure 2-14. An example of solving DFT through HDFT method

The following issue most be considered in the implementation of this procedure for decreasing the volume of computations. Dynamic gates are separated from static gate. This is because solving static gates does not need SMP theorem. They can be easily solved by the use of conventional probability theorem. The solution of first layer gates can be easily carried out if their inputs have exponential CDF because gates with inputs with

exponential CDF are solved with Markov theorem rather than SMP theorem. This consideration decreases the volume of computations. Based on these issues the flow chart of SMP-based DFT's reliability solution will be as Figure 2-15.

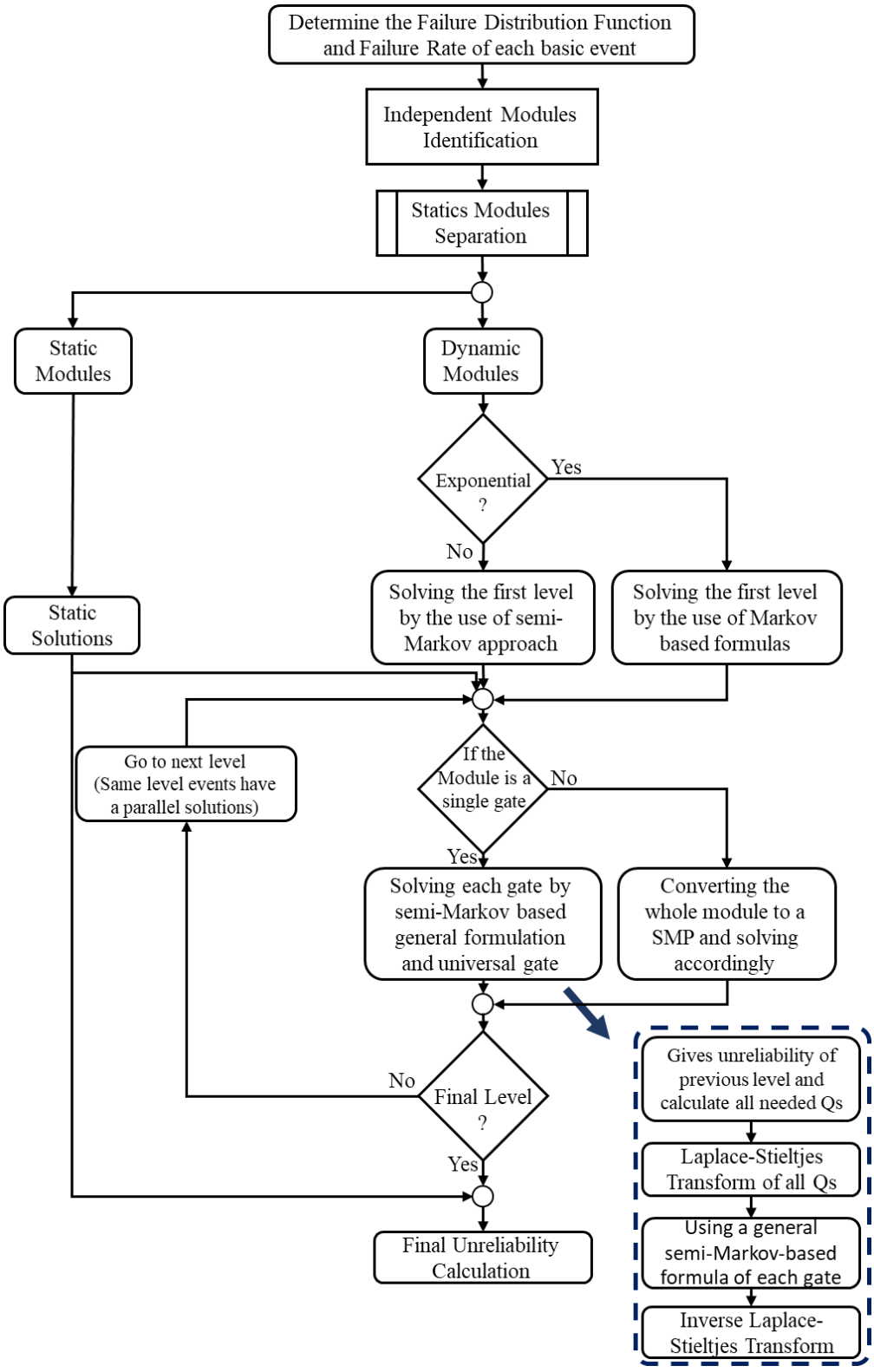


Figure 2-15. Flow chart of SMP-based DFT's reliability solution (the proposed method of this project)

2.11 Results

In this section, six examples are given to indicate the capabilities of the HDFT method. The first example makes a comparison between the results of the HDFT method and those obtained from the Windchill Quality Solution (WQS) software. The challenge point in this example is precision of the result in comparison with commercial tools. The second example studies the ability of this method applied in a DFT with repeated events. So, the challenge point of this example is about dealing with repeated events. In the third example, we will apply the HDFT method on HCAS and compare the maximum number of states and transitions in three selected Markov-based DFT's solution approach. The robustness to state and transition explosion will be the challenge point of this example. The fourth example solves a selected DFT with event with exponential and non-exponential CDF, and then compares the results taken from the HDFT method, and algebraic technique. Dealing with non-exponential CDF can be a challenge in reliability evaluation that is considered in example four. As the repair consideration can be a challenging issue in DFT, the fifth example discusses on the possibility of using HDFT method for repairable DFTs. The final example deals with a case study of Aircraft Fuel Distribution System (AFDS).

2.11.1 Example of How This Method Works?

In order to verify the accuracy of this method and discuss the way, it works a simple example of DFT including a cold spare gate (CSP) and a PAND gate is provided as Figure 2-16.

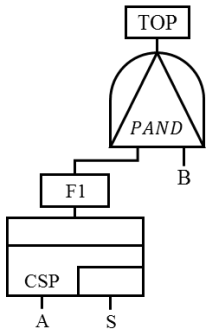


Figure 2-16. Example 1 (scenario 1): a tree with two dynamic PAND and CSP gates and three basic events

The CTMC for the above DFT is illustrated in the following figure. This model contains seven states (two absorbing states and five operational states). In this example, the reliability achieved through the HDFT method is compared with the one achieved

through Markov model, so that the abilities of the HDFT method and its limitation can be discussed.

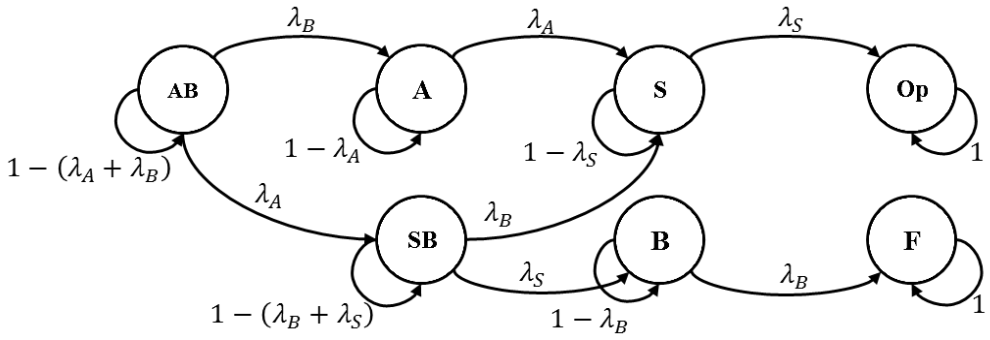


Figure 2-17. The converted Markov model for reliability evaluation of the DFT in Example 1 (scenario 1).

The above model can be redrawn as a semi-Markov model which is illustrated in the next figure. This model is extended to improve the ability of model in considering any failure distribution function.

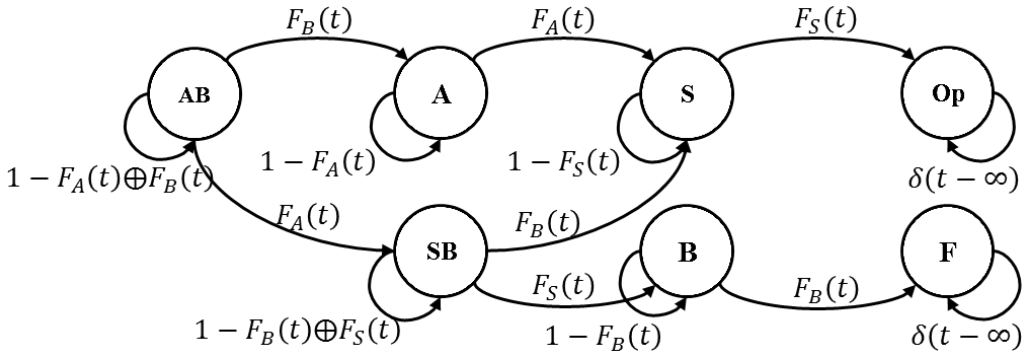


Figure 2-18. The converted SMP model for reliability evaluation of DFT in the Example 1 (scenario 1).

If the CSP is solved individually, and the achieved expression from its solution is imported to the semi-Markov model (the HDFT method of this thesis), then semi-Markov model for above-mentioned DFT of the first example can be illustrated in the following figure.

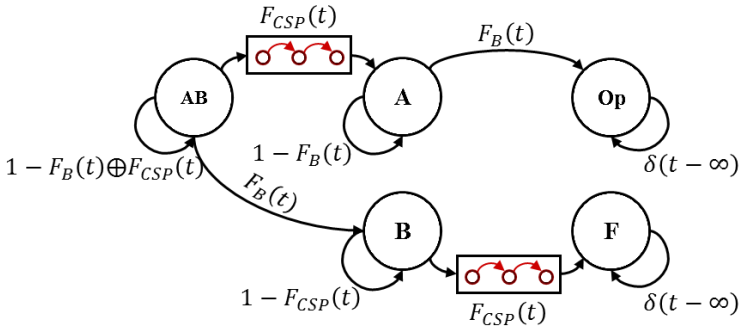


Figure 2-19. Hierarchical semi-Markov model (the HDFT approach) for reliability evaluation of DFT in Example 1 (scenario 1).

When the DFT is hierarchically modelled with semi-Markov model (HDFT method), some transitions between lower levels' states are omitted. In other words, the displayed transition in the following figure (red transition) is omitted in the HDFT method of this thesis. Since this transition does not create a path to the system failure, according to Markov Path Theorem in reliability solutions, it cannot have any effect on the probability of system failure. Hence, systems' solution through HDFT method and Markov model should be equal. Note that, subsidiary transition is a transition which transits from a state of sub-Markov in box to a state of the other sub-Markov in another box.

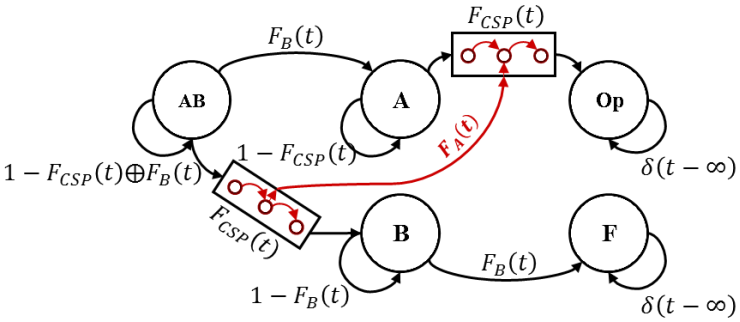


Figure 2-20. Hierarchical semi-Markov model (without omitting the subsidiary transition) converted for reliability evaluation of DFT in Example 1 (scenario 1)

In order to solve this example, it is assumed that modules' failure obeys exponential distribution function and failure rates of the events are $\lambda_A = 0.06$, $\lambda_B = 0.07$, $\lambda_S = 0.08$. The failure rates are scaled by failure per hour. Considering 0-100 hours for mission time, the reliability results using CTMC, and the HDFT method (hierarchical DFT solution using Semi-Markov Process-based approach) can be illustrated as Figure 2-21. It can be seen that both achieved reliabilities match one another and calculation error is zero.

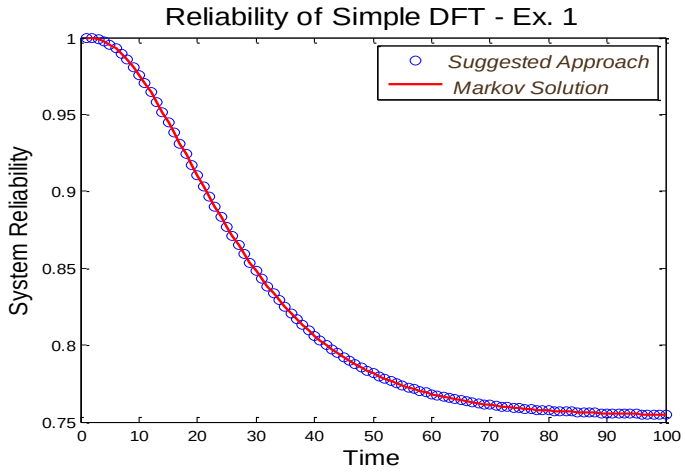


Figure 2-21. Reliability of the illustrated DFT in example 1, scenario 1 through CTMC and the HDFT method.

If we locate the CSP in the second input of the PAND gate, the DFT would be in form of Figure 2-22. This tree has no difference with the previous one in terms of complexity, however, the outputs of two methods are different which are discussed in the following.

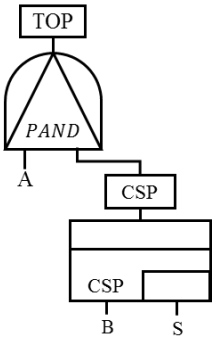


Figure 2-22. Example 1 (scenario 2), using CSP in the right-side input and the basic event in the left-side.

When a gate such as spare gate (or gates that are nature of redundancy and masking the failures) is located in the right-side of PAND gate, a subsidiary transition would be omitted in the proposed method as in Figure 2-23 which is influential in system failure. In the other words, one of the failure paths is omitted. Therefore, the answer is closer to the precise analytical answer, but it is not the same. In addition, when the redundancy is applied more, the achieved answer in comparison to the correct one is more inaccurate.

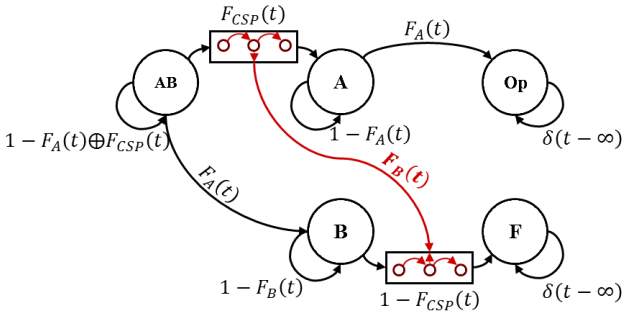


Figure 2-23. Hierarchical semi-Markov model (the proposed method) for reliability evaluation of DFT in example 1, scenario 2. Note that the subsidiary transition is omitted in the proposed method.

Figure 2-23 shows the reliability results through CTMC solution and the proposed method in this thesis for DFT in example 1, scenario 2. As it is previously said, due to the employed proximity in the proposed method and due to the omission of subsidiary transition of the achieved reliability in the proposed method, it is a bit higher than the real amount.

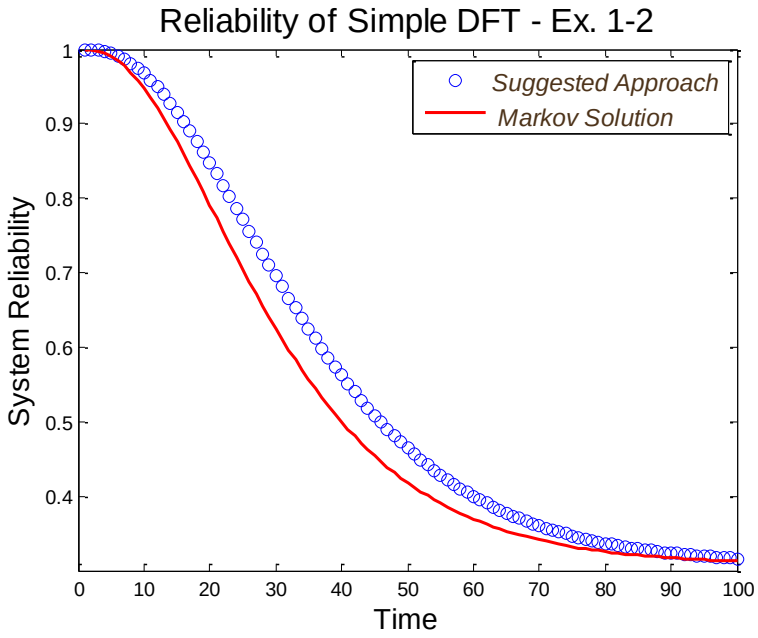


Figure 2-24. Reliability of the illustrated DFT in Figure 2-22 through CTMC and the proposed method of this thesis

The third scenario of this example is to add two CSP into both inputs of the PAND gate. In this case, the system DFT is illustrated in the following figure. In this model, it is assumed that the failure rate of both spares is 0.08 failure per hour and A and B input failure rates are assigned like before. It should be noted that in DFT, gates and events are given numbers from top to bottom and from left to right. In addition, the left CSP is called CSP1, whereas the other one is called CSP2.

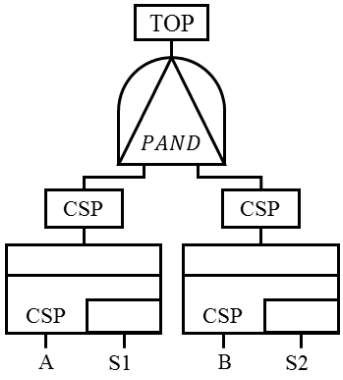


Figure 2-25. Example 1 (scenario 3), using CSP in both PAND gate's inputs.

The semi-Markov model for the above DFT is illustrated as follows. As it can be seen in this model, there are two subsidiary paths and one subsidiary state which are omitted during the use of the proposed method of this thesis, so this causes the achieved reliability to be higher or lower than the real amount.

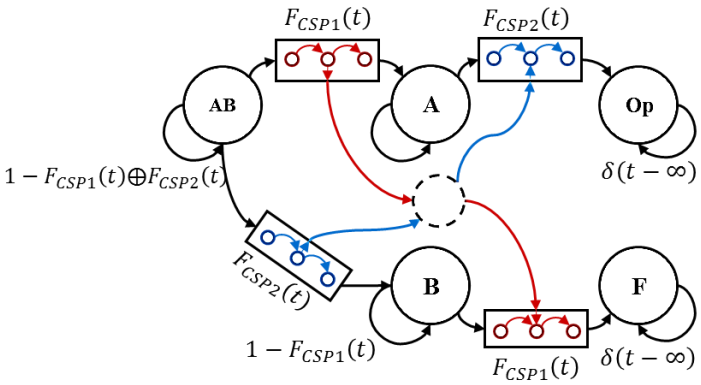


Figure 2-26. The hierarchical semi-Markov model (the proposed method) for evaluating the DFT reliability in example 1, scenario 3

As it can be seen in the above figure, while using the proposed method, the subsidiary transitions are omitted as well as the subsidiary state. The reliability diagram of the mentioned DFT is also illustrated in the following figure comparing CTMC solution and the proposed method. It can be seen in this method that the achieved answer from the proposed method is higher than the exact answer. But it should be noted that, in this case, the state difference between the answers is lower than the previous state. In other words, in the second scenario the second difference norm of the answers is 0.40042 and in the third scenario, the second difference norm of the answers is 0.17949. According to the experiences in this example and special features of CTMC of each dynamic gates, it can be said that whenever AND, PAND and SPARE gates are located in the left side input of a gate same as themselves, the answer achieved from the proposed method is equal to the exact answer. But if these gates are located in the right-side input or both inputs of a gate the same as themselves, the answer is a bit higher than the analytical answer in terms of reliability which totally depends on the number of gates in the previous level and their failure probability. Meanwhile, the proposed method of this thesis has no limitation for SEQ, CSP, OR and FDEP gates and the DFT which possesses these kinds of gates are solvable regardless of their inputs in the previous levels (i.e., in this state, the inputs can be AND, PAND and SPARE gates). It should be noted that there is an exception in case of PAND gates which means if this gates' input is PAND, the final answer through the proposed method is equal with the exact answer achieved through CTMC solution.

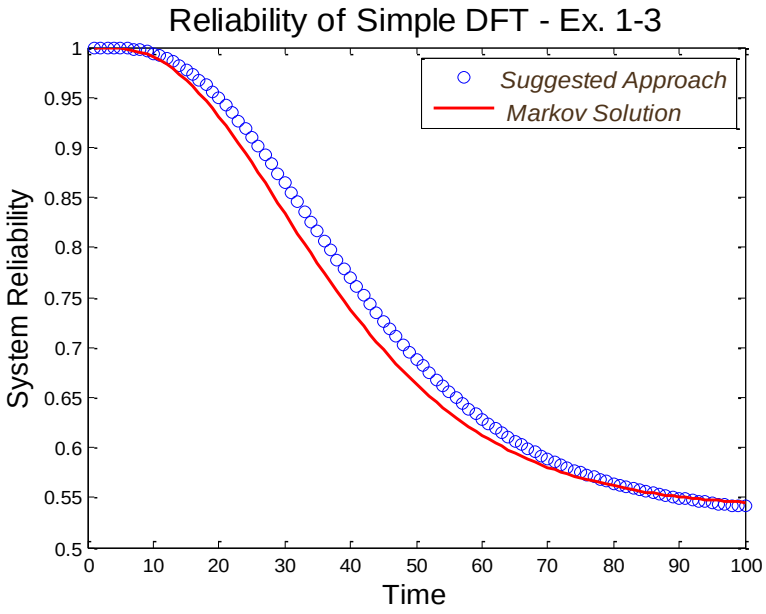


Figure 2-27. Reliability analysis of example 1 – scenario 3.

2.11.2 Example 2. DFT with a PAND Gate and Static Gates

From the DFT shown in the following figure. it can be seen that this DFT consists of a PAND gate, two static gates and 10 basic events. Solving this DTF by the use of Markov theorem requires solving a CTMC with 160 states (Zhu, Han, Liu, & Zuo, 2014; Amari, Dill, & Howald, 2003). In contrast, the proposed method can solve this DFT through SMP theorem with only 5 states, it can even provide the parametric expression for reliability. Metrics such as sensitivity and MTBF can easily be evaluated through parametric results of the proposed method.

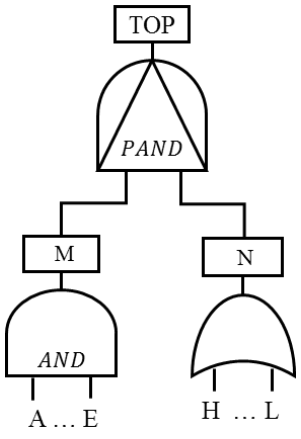


Figure 2-28. DFT of example 2. A tree with PAND gate, two static gates and 10 basic events (Zhu, Han, Liu, & Zuo, 2014).

The failure rates of the above DFT's events have been listed in the following table. In this table, failure rates are "failure per hour".

Table 2-1. Failure rates of the above DFT (Amari, Dill, & Howald, 2003)

Basic Events	Failure Rates	Basic Events	Failure Rates
λ_A	0.0011	λ_H	0.011
λ_B	0.0012	λ_I	0.012
λ_C	0.0013	λ_J	0.013
λ_D	0.0014	λ_K	0.014
λ_E	0.0015	λ_L	0.015

According to the result obtained in reference (Amari, Dill, & Howald, 2003), the amount of unreliability of the system in 1000 hours of the mission time is 0.363. The value obtained from our proposed method (0.363024069761471) is completely coinciding with this result.

2.11.3 Example 3. DFT with Repeated Basic Event

This example examines the capability of the proposed method to model a DFT with repeated events. To do this the DFT of references (Yuge & Yanagi, 2008; Zhu, Han, Liu, & Zuo, 2014) which is shown in in the following figure is selected.

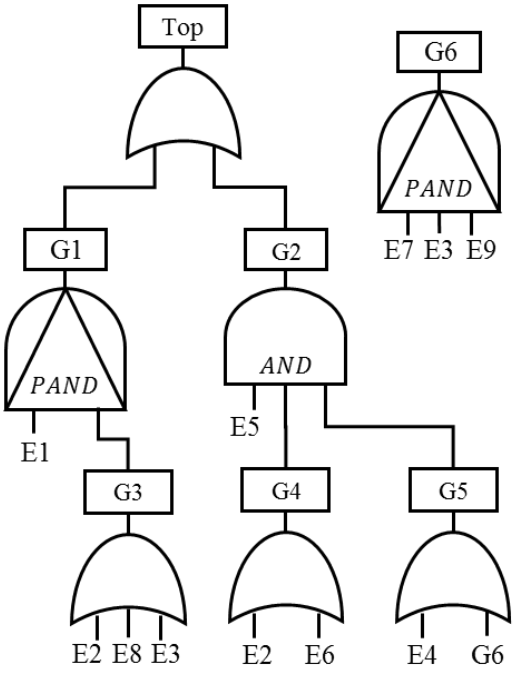


Figure 2-29. DFT of example 2. A DFT with the repeated event (Yuge & Yanagi, 2008)

This DFT consist of nine basic events in which the event E2 is repeated. The failure rate of all basic events is set 0.01 failure per hour. Solving the above DFT in the time interval 0-300 hours has been provided through three methods in reference (Yuge & Yanagi, 2008), using Galileo software, Monte Carlo simulation and a Yuge's method. Of these methods Galileo a more exact solution because its underlying approximation in the

solution process of DFT is less than the others. This is why research works like (Yuge & Yanagi, 2008) take the outcomes of Galileo as a reference.

We solve the mentioned DFT through our proposed method, and the results are given along the results from the considered reference. The accuracy and precision of our method are then analysed. The following table indicates the results. As seen the results of our method is closer to those of Galileo's results than other two methods (Yuge's method and Monte Carlo simulation).

Table 2-2. Comparing the results obtained from solving DFT in the previous figure of reference (Yuge & Yanagi, 2008) and the proposed method of this thesis.

Time	Galileo	Yuge's Method	Monte Carlo	Proposed method
0	0.00000	0.00000	0.00000	0.00000
50	0.21418	0.25653	0.25647	0.19526
100	0.49318	0.59960	0.59970	0.45148
150	0.68751	0.80196	0.80212	0.64738
200	0.81010	0.90114	0.90120	0.78226
250	0.88519	0.94864	0.94869	0.86873
300	0.93066	0.97213	0.97215	0.92168

The following figure shows the results graphically and as can be seen the proposed approach has a close reliability value to the ground truth that is generated by Galileo software. In addition, the provided solution has a better approximation of the reliability in comparison to the Monte Carlo simulation (which is time consuming).

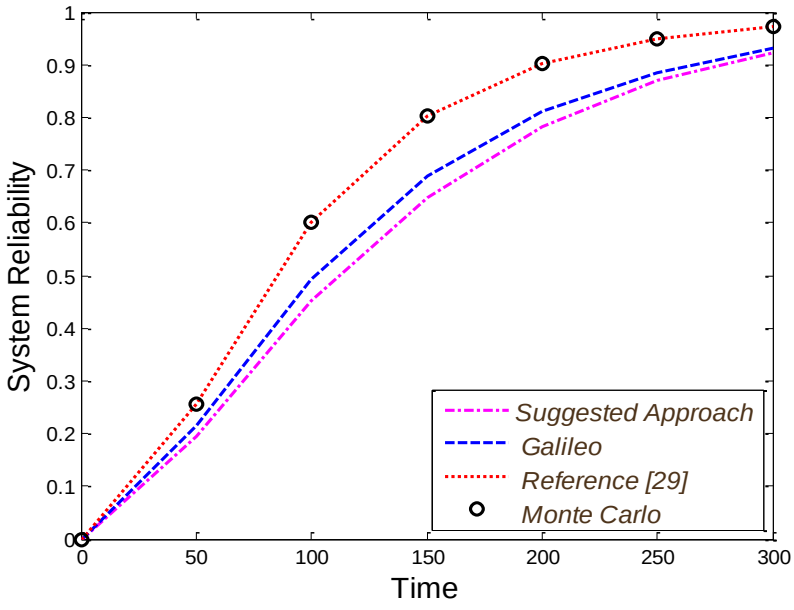


Figure 2-30. The unreliability of the DFT in example 3 and comparing the results of the proposed method of this thesis with results from reference (Yuge & Yanagi, 2008).

The following table indicates the results in terms of norm function. In this table, Q_S stands for the unreliability results obtained from Monte Carlo simulation. Q_G is the

unreliability results obtained from Galileo and Q_P denotes the unreliability results obtained from our proposed method. $\|Q_P - Q_G\|$ shows the difference between the unreliability of our proposed method and Galileo software.

Table 2-3. Comparing difference norm of the results in reliability solution of DFT (example 3)

	1-norm	2-norm	infinity-norm
$\|Q_S - Q_G\|$	0.45951	0.20080	0.11461
$\|Q_R - Q_G\|$	0.45918	0.20062	0.11445
$\|Q_P - Q_G\|$	0.15401	0.06951	0.04169

The above table shows the superiority of the proposed method (in this thesis) to the others.

2.11.4 Example 4. DFT of HCAS

Hypothetical Cardiac Assist System (HCAS) is a benchmark in validating different methods applied to DFT reliability solution which has also been discussed in various references. The DFT of HCAS as shown in the following figure, including CPU module, motors module, and pumps module. This DFT consists of a shared CSP in the pump section.

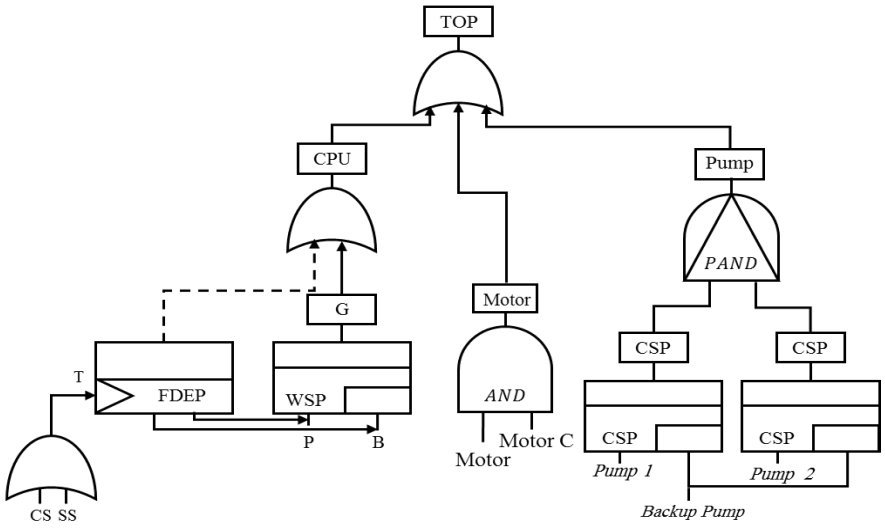


Figure 2-31. DFT of HCAS with shared CSP gates in the pump module

The failure rates of basic events for this DFT have been assigned in the following Table. Reference (Boudali & Bechta Dugan, A Discrete-Time Bayesian Network Reliability Modeling and Analysis Framework, 2005) has solved this DFT based on Bayesian Networks for 100000 mission time, and the system unreliability obtained 0.36501.

Table 2-4. Failure rates of basic events in DFT of HCAS in example 4 (Boudali & Bechta Dugan, A Discrete-Time Bayesian Network Reliability Modeling and Analysis Framework, 2005)

Basic Events	Failure Rates $(10^{-6}$ F/hr$)$	Basic Events	Failure Rates $(10^{-6}$ F/hr$)$
CS	1	$Pump_1$	5
SS	2	$Pump_2$	5
P	4	$Pump_{Backup}$	5
B	4	$Motor$	5
$Motor_C$	1		

The unreliability of this system computed from our proposed method is 0.363500847376541 which is more precise and in agreement with other research works' results. Another significant point in solving this model using the proposed method is decreasing the size of the corresponding Markov model of these DFTs. Different strategies may lead to generating different state-space model. To clarify this, consider the following table. This table presents the number of states and the number of transitions of the corresponding Markov model of the largest module in DFT when solving by DIFtree, Coral and proposed method. The detail on how the Markov model is generated can be found in (Bavuso *et al.*, 1994; Dugan, Venkataraman and Gulati, 1997). As can be seen from the following table, our proposed method possesses the lowest number of states and transitions. Generally speaking, the proposed method enables to decrease the number of states by the use of a reshaped model with load sharing gate.

Table 2-5. A comparison between different Markov-based methods for DFT solution of HCAS (Boudali & Bechta Dugan, A Discrete-Time Bayesian Network Reliability Modeling and Analysis Framework, 2005; Boudali, Crouzen, & Stoelinga, A Rigorous, Compositional, and Extensible Framework for Dynamic Fault Tree Analysis, 2010)

Analysis Method	Max. States Number	Max. Transitions Number
DIFtree or Galileo	8	10
Coral	36	119
Proposed method	5	5
Proposed method with Markov-Path	3	2

As shown in the above figure, the DFT has eight gates and based on the proposed algorithm in each iteration only one gate will be solved through SMP, and the CDF of the result will be stored to be used as an input for the next level gates. It means in each iteration the maximum number of states will never be higher than five based on the provided universal gate. Similarly, the maximum number of state transitions will never be higher than five as well. In other words, in HCAS DFT, there are two PANDs, one LSH, one AND, one WSP, one FDEP and three OR which means the maximum number

of states will be used for evaluation of PAND gate. In addition, the maximum number of state transition will be used for evaluation WSP with FDEP on its.

The computational complexity of the Semi-Markov Reward Processes has been studied by (Ciardo, Marie, Sericola, & Trivedi, 1990) and if we consider the reward zero for operational states and reward one for failed state(s), the computational complexity of SMPs can be described as $O((x^{[00]}+x^{[01]})n)$ where n is the number of states, $x^{[00]}$ is the number of nonzero transitions between operational states and $x^{[01]}$ is the number of nonzero transitions from operational states to failed state(s). The proposed approach has a loop that evaluates the SMPs for each gate in the DFT. Thus, the computational complexity of the proposed method will be $O((x^{[00]}+x^{[01])}$ k n) where k is the number of gates in the fault tree. Based on the proposed universal gate, the maximum value of $x^{[00]}+x^{[01]}$ can be five and the maximum number of state can be also 5. Therefore, because of bounded value in both number of states and number of transitions, the computational complexity of the proposed method is only depending on the number of iterations and can be simplified in O(k). On the other hand, the computational complexity of CTMC can be described as $O(n^2)$ (Ciardo, Marie, Sericola, & Trivedi, 1990) where n is the total number states. It is clear that the proposed approach has less computational complexity as it was expected than a traditional CTMC based approach.

2.11.5 Example 5. DFT with PAND gate and non-exponential failure

This example contains events with non-exponential CDF (Weibull CDF) to show the capability of the proposed method. The CDF of Weibull can be described by (2-42).

$$F(t) = 1 - e^{-(t/\omega)^{\sigma}} \quad (2\text{-}42)$$

where σ is the shape and ω denotes the scale parameter of the Weibull CDF. First, consider the DFT of Figure 2-28.

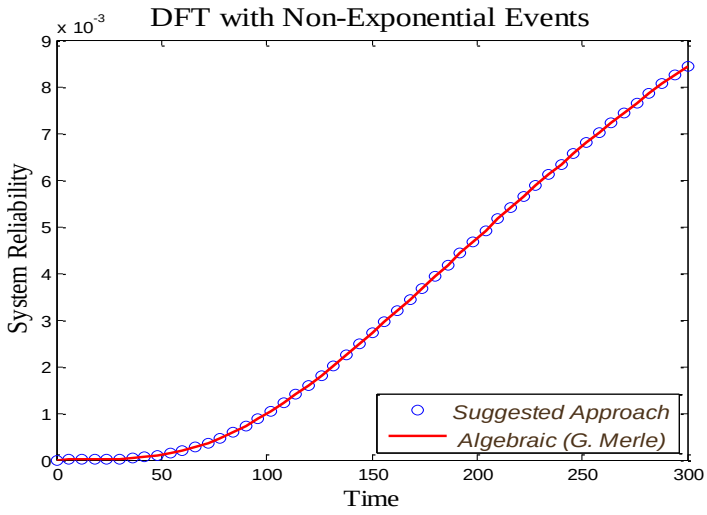


Figure 2-32. Unreliability evaluation of the DFT with PAND gate and non-exponential failure distribution in its BEs.

The failure rate of basic events in this tree obeys exponential CDF with the values of Table 2-1 values except for J, K, and L. It is assumed that the failure distribution of J, K, and L obeys Weibull CDF and the value of σ is 0.1 and ω is 20. The following shows the reliability of this DFT. As seen from this figure, both methods (proposed method and Algebraic method) give exactly the same results.

2.11.6 Example 6. Repairable DFT

The first generation of DFT was unable to repair modeling. References (Codetta-Raiteri D. , 2011; Codetta-Raiteri D. , 2005; Portinale, Raiteri, & Montani, 2010) presented an extended DFT able to model repair actions and solved it by the use of Generalized Stochastic Petri Net (GSPN) and Dynamic Bayesian Networks (DBN). This thesis addresses the modelling of repair gate by a hypothetical example. It should be noted that in the current state we cannot guarantee that the proposed approach will work for any other repairable DFTs. In fact, this example is only a sign that there might be an opportunity to improve the proposed approach for repairable DFTs. The complete development of this modelling remains as future research work.

Through an example, we show that this modeling method is a good solution for finding the reliability of repairable DFT. The following figure illustrates a repairable DFT with failure and repair rates of the following table.

Table 2-6. Failure rates and repair rates of basic events in repairable DFT of example 6 – event C is a non-repairable event.

Basic Events	Failure Rates (F/hr.)	Basic Events	Repair Rates (R/hr.)
λ_A	0.001	μ_A	0.01
λ_B	0.002	μ_B	0.01
λ_C	0.003	μ_C	0.00

Repairable DFT is shown in the following figure illustrates that in the case of finding the output of AND gate, events A and B can be repaired. In this system, it is assumed that event C is non-reparable.

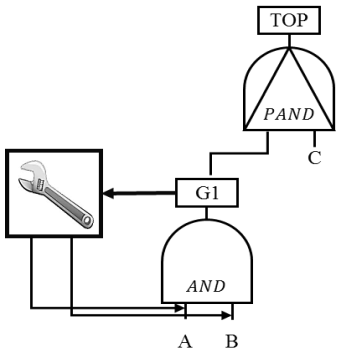


Figure 2-33. A hypothetical repairable DFT with a repair gate.

The CTMC of this tree is illustrated in the following figure. It is assumed that the repair actions carry out immediately after failing the related events. This issue has been shown in the figure by dash transitions with repair rate μ. In order to avoid model complexity in this figure, some self-transitions are not depicted.

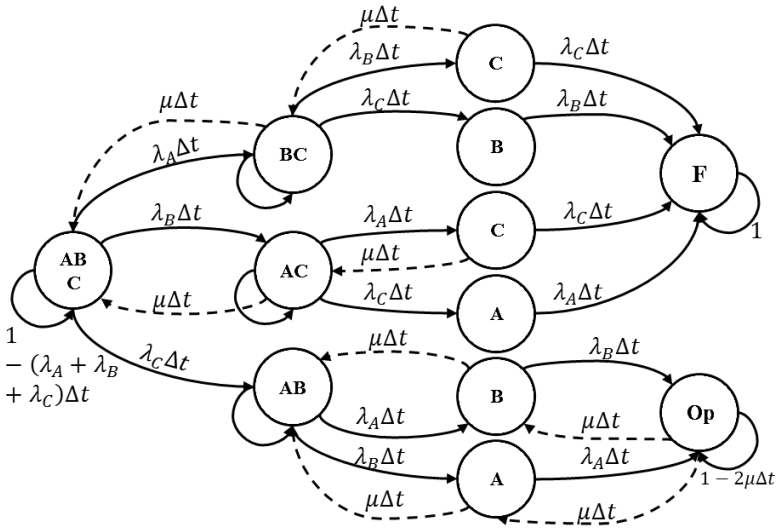


Figure 2-34. CTMC of the repairable DFT of Figure 2-33

Recall that our hierarchical method solved the given DFT layer by layer in a bottom-top manner. For examples the DFT of Figure 2-33 consists of two layers. First, the bottom layer (AND gate with repairable events; AND and repair gates) is solved. Its output is delivered to the left-hand-side input of PAND gate (up to a layer of the DFT). The semi-Markov model of this gate with an input for which the impact of repair has been considered is shown in the following figure. The dashed transitions defer the semi-Markov model of the PAND gate with repairable events from a simple PAND gate (a

PAND gate with non-repairable events). The general equation of this gate is achieved through solving tits semi-Markov model by SMP theorem as follows. It should be noted that in this model G represents repair distribution function of the input's components which is equal to $G=1-e^{-\mu t}$, if repair distribution is considered exponential.

$$R(t)=1-L^{-1}\left\{\frac{q_{1,2}(s)q_{2,4}(s)}{1-q_{1,2}(s)q_{2,1}(s)}\right\} \tag{2-43}$$

where the $q_{i,j}(s)$ is the LST form of $Q_{i,j}(t)$ explained in the methodology section.

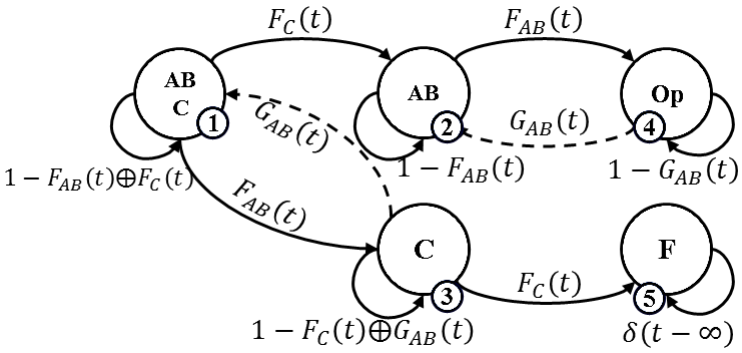


Figure 2-35. Semi-Markov model for PAND gate with reparability in the first input

This procedure for solving DFT with repair gate (gates) can be applied for any other gates. This work is an open research direction for. Regarding both two presented models (CTMC and semi-Markov models), the unreliability behaviour of the supposed DFT will be as the following figure for 700 hours mission time. As seen from this figure, our method gives results very close to those achieved by the CTMC method. This verifies the correctness of our method.

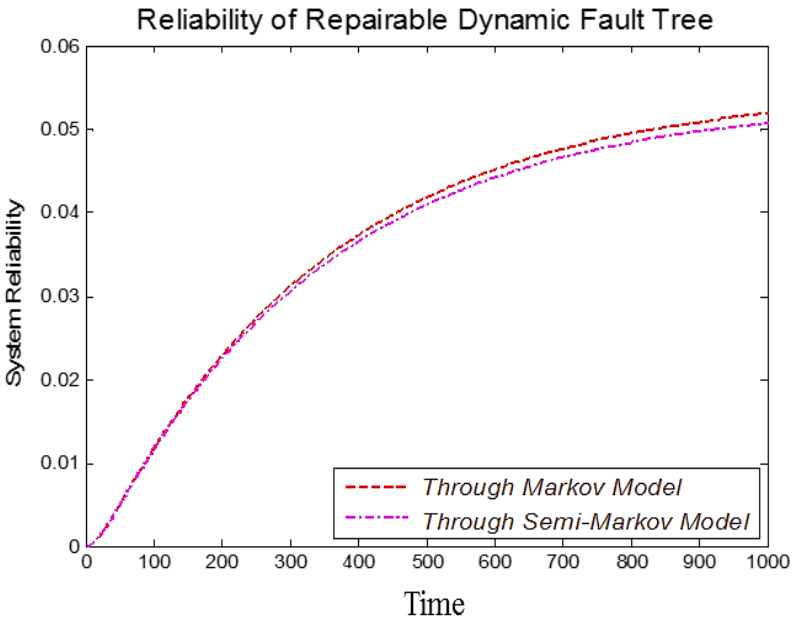


Figure 2-36. The unreliability of repairable DFT in two states of solving by CTMC and solving by SMP theorem

2.11.7 Example 7. Aircraft Fuel Distribution System

As a case study, the Aircraft Fuel Distribution System (AFDS) has been chosen from (Edifor, Gordon, Walker, & Papadopoulos, 2014) and its DFT from (Kabir, Walker, & Papadopoulos, 2018). The following figure illustrates the schematic of AFDS and Figure 2-38 shows the DFT of the system that has been derived by HIP-HOPS tool (Papadopoulos, Mcdermid, Sasse, & Heiner, 2001; Kabir, Walker, & Papadopoulos, 2018).

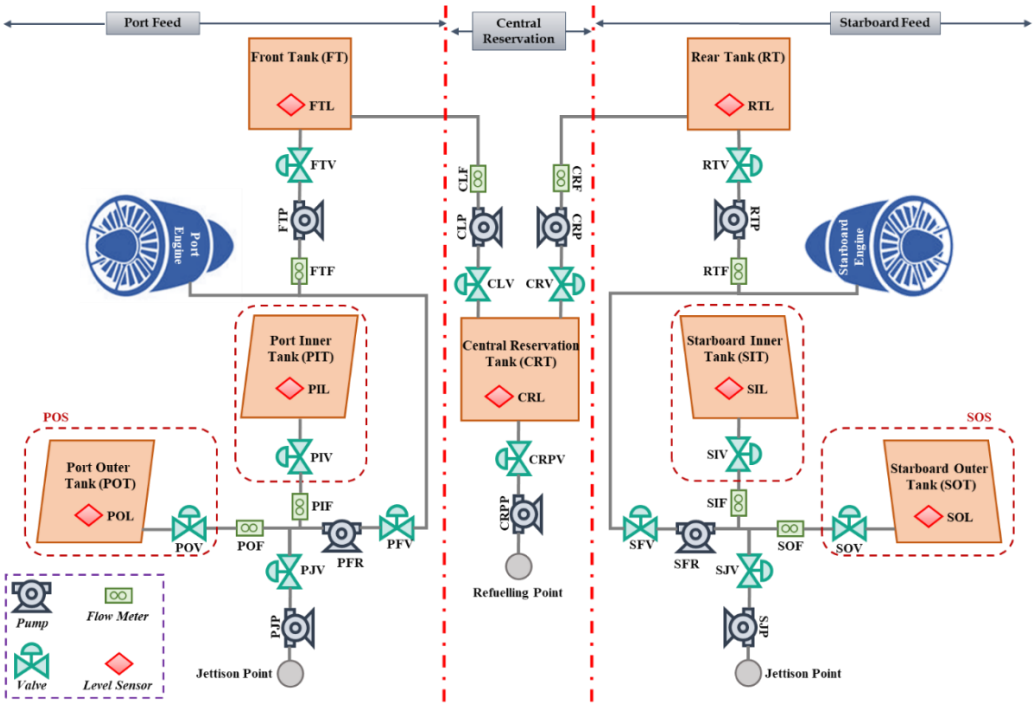


Figure 2-37. Schematic of Aircraft Fuel Distribution System (Modified from (Edifor, Gordon, Walker, & Papadopoulos, 2014))

This system has two engines, seven bi-directional fuel pumps, five fuel tanks, and eleven valves enables the control system to choose active paths for fuel distribution in different conditions. The system also has six flow meters for fuel flow rate measurement. To refill the tanks, there is a refuelling point and there are two jettison points for releasing the fuel in some situations.

The AFDS has been divided into three parts for performing the compositional analysis including Starboard Feed (SF), Central Reservation (CR) and Port Feed (PF) as delineated in the above figure. As can be seen, two SF and PF have identical components and each one of them has some subsystems. For instance, the SF includes Starboard Inner Subsystem (SIS), and Starboard Outer Subsystem (SOS). Through further decomposition of those subsystems, we have some components. For example, the SIS contains Starboard Inner fuel Level sensor (SIL), the valve (SIV) and Tank (SIT).

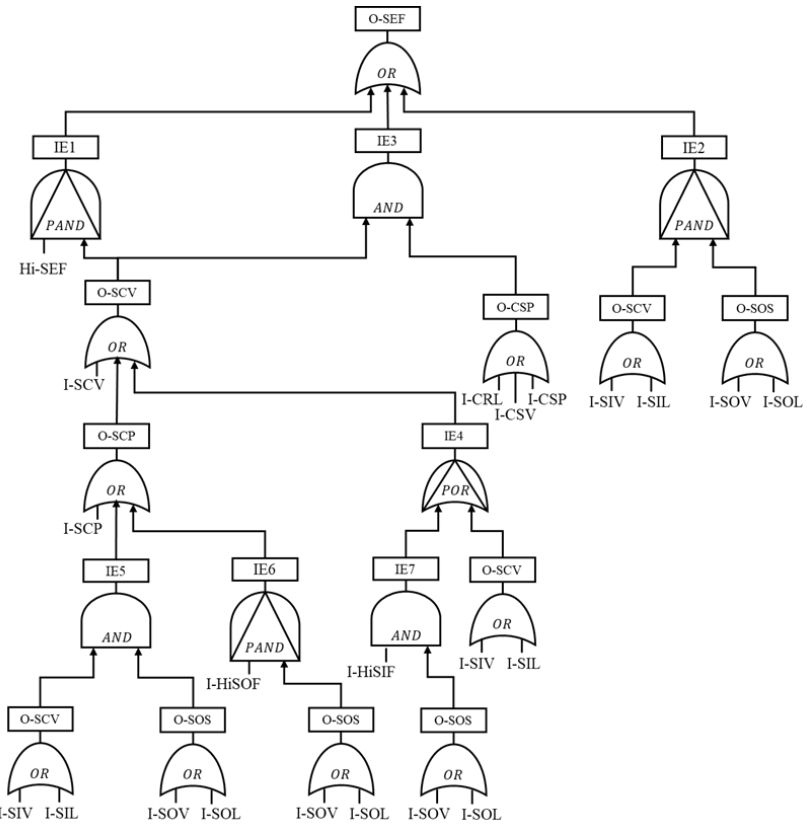


Figure 2-38. Dynamic Fault Tree of Aircraft Fuel Distribution System (Kabir, Walker, & Papadopoulos, 2018).

Distribution of the fuel throughout the system and storing the fuel in the thanks are two main functions of AFDS and each function can be divided into two phases of refuelling and consumption for different situations such as taxiing, take-off, cruising, approaching, and landing.

The fuel will be injected into Central Reservation Tank (CRT) in the refuelling phase and then automatically distributed to Starboard and Port tanks. Moreover, the fuel will be consumed by both Starboard and Port engines in the consumption phase and certain level of fuel will be fed to engines. For more details regarding the ADFS please read (Kabir, Walker, & Papadopoulos, 2018).

In DFT of AFDS, It should be noted that “O- CompX“ stands for omission of functionality of component X, “I-CompX” refers to the internal failure of component X and “Hi-CompX” includes erroneous high reading from component X.

The DFT of AFDS has twelve identical basic events (twenty-two basic events counting the repeated events). The following table provides the failure rates and also short descriptions for the basic events.

Table 2-7. Failure rates and basic events in DFT of AFDS (Kabir, Walker, & Papadopoulos, 2018)

Basic Events	Failure Rates (F/hr.)	Descriptions
I-SCP	5.84267E-5	Internal Failure of Starboard-Central Pump
I-CSP	5.84267E-5	Internal Failure of Central-Starboard Pump
I-SOV	1.65633E-3	Internal Failure of Starboard Outer Valve
I-SIV	1.65633E-3	Internal Failure of Starboard Inner Valve
I-CSV	1.65633E-3	Internal Failure of Starboard Central-Starboard Valve
I-SCV	1.65633E-3	Internal Failure of Starboard Starboard-Central Valve
I-CRL	2.21127E-6	Internal Failure of Central Reservation Level Sensor
I-HiSOF	4.06861E-5	Internal Failure or High Reading from Starboard Outer Feed
I-HiSIF	4.06861E-5	Internal Failure or High Reading from Starboard Inner Feed
I-HiSEF	4.06861E-5	Internal Failure or High Reading from Starboard Engine Feed
I-SIL	1.65633E-3	Internal Failure of Starboard Inner Level Sensor
I-SOL	3.31774E-5	Internal Failure of Starboard Outer Level Sensor

Reliability evaluation of AFDS through DFT has been addressed by (Kabir, Walker, & Papadopoulos, 2018). They have converted the DFT to Petri Nets and Bayesian Networks and consequently calculated the reliability of the system. A comparison between the proposed approach and two other existing addressed methods has been provided in the following table. The obtained results are approximate but close to the existing results.

Table 2-8. Comparison between proposed approach, Petri Nets-based, and Bayesian Networks based approaches.

Mission Time	Proposed Approach	Petri Nets (Kabir, Walker, & Papadopoulos, 2018)	Bayesian Networks (Kabir, Walker, & Papadopoulos, 2018)
100	0.05261416997	0.04998134610	0.04527486000
500	0.59015071681	0.55645041000	0.52900833000
1000	0.89056414810	0.87518982420	0.85004877000
1500	0.96590195305	0.96217232090	0.94597227200
2000	0.98766528020	0.98690630030	0.97782727200
2500	0.99514980424	0.99502308600	0.99029561190
3000	0.99801543278	0.99800983670	0.99563683150
3500	0.99917346098	0.99918448590	0.99809093140
4000	0.99965314139	0.99966205560	0.99916055205
4500	0.99985398520	0.99985924100	0.99960976330
5000	0.99993845915	0.99994123190	0.99980478790

Regarding the reliability evaluation of DFTs, there are some other challenges such as Common Cause Failures (CCF), and Reconfiguration that are not considered in this thesis and can be studied as the future research.

2.11.8 Example 8. Incorporating Complex Basic Events

To illustrate the idea of safety analysis of systems with complex BEs via the proposed approach, we use a simplified version of the oxygen sensing and generation unit of an Automatic Pond Oxygen Management System and shown in the following figure. The role of this system is to continuously sense the oxygen level of a pond and if the oxygen level falls below a certain level, then the system will automatically generate oxygen.

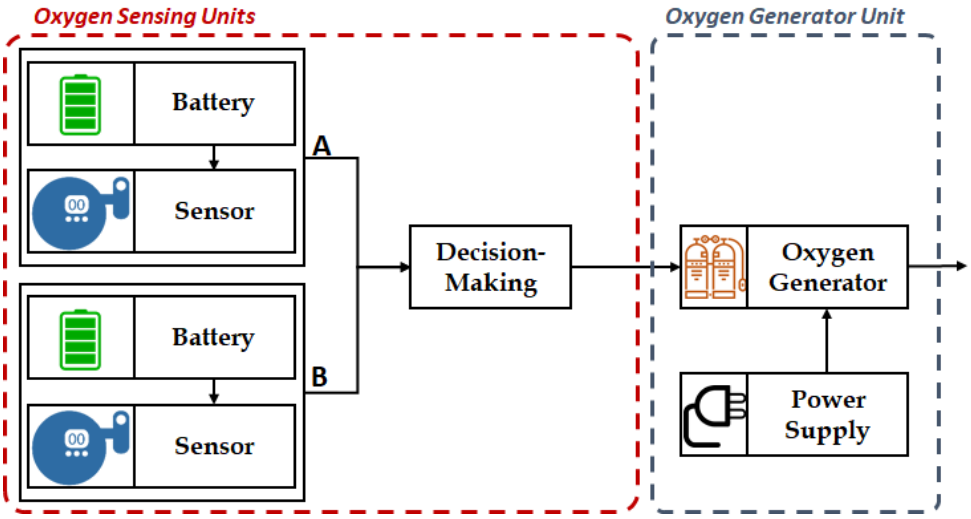


Figure 2-39. Automatic Pond Oxygen Management System.

The system contains two oxygen level sensing blocks, A and B. Each of these blocks contains a battery and an oxygen sensor. The battery keeps the sensor alive, and the sensor senses the pond's oxygen level. Readings from both blocks are fed to the Decision Making (DM) block. Based on these readings, the DM can decide whether to generate oxygen or not. Note that although both block A and B work simultaneously, input from at least one of them is necessary to be decided by the DM. When the DM finds that it is necessary to generate oxygen, it uses the oxygen generator (OG) unit to generate oxygen. During operation the OG draws power from the power supply.

For a model-based analysis of this system, the architecture of the system was annotated by considering the failure behaviour of each of the system components. A fault tree was automatically generated based on this annotated architecture and shown in the following figure.

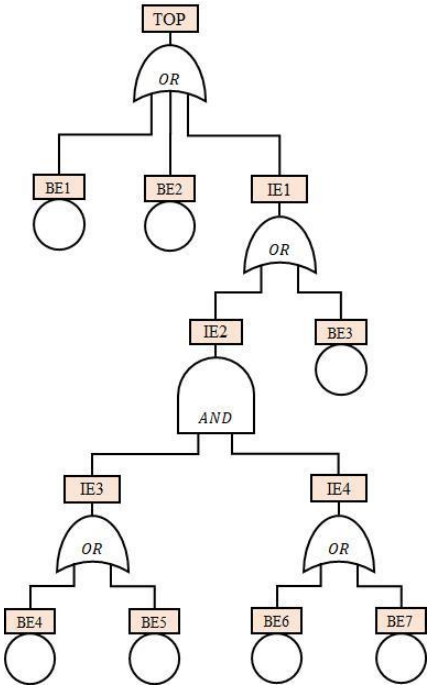


Figure 2-40. Fault Tree of Automatic Pond Oxygen Management System

Table 2-9 shows the ID and description of the basic and intermediate events of the fault tree. In this study, basic events 1, 4, and 6 were considered as complex basic events. The SMP-based failure behaviour models of these BEs are shown in Figure 2-41. Parameters associated with these models and failure rates of other BEs are shown in Table 2-10.

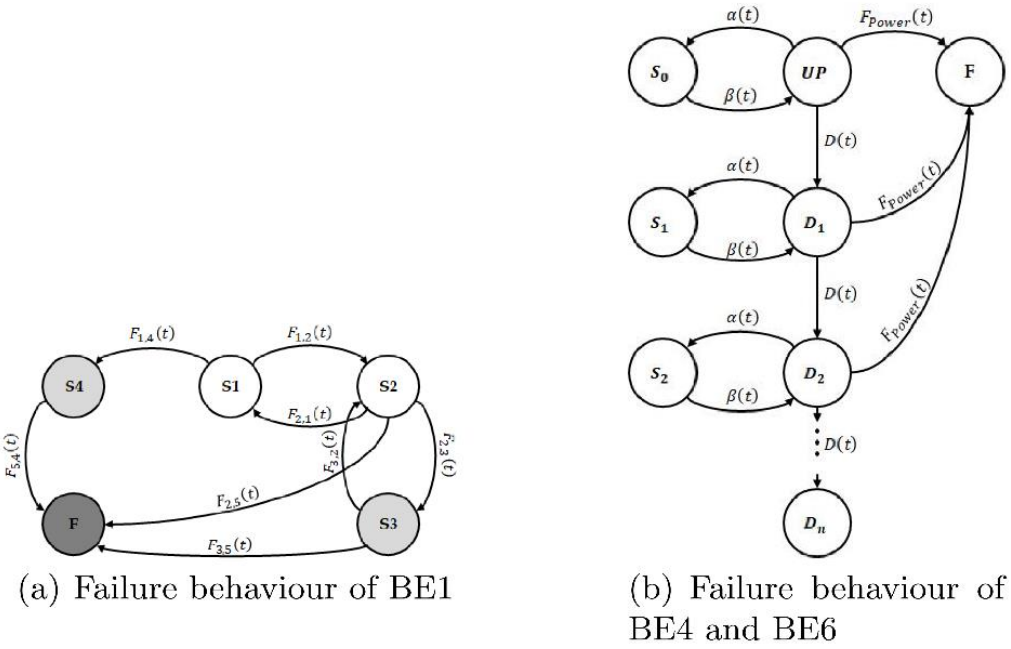


Figure 2-41. (a) Failure behaviour of BE1 (b) Failure behaviour of BE4 and BE6 (Kabir, Aslansefat, *et al.*, 2019).

Table 2-9. ID and description of the basic and intermediate events of the fault tree shown above.

Event ID	Event Description
TE	No oxygen generated when required
IE1	No outputs from decision making block
IE2	No output from oxygen level sensing blocks
IE3	No output from oxygen level sensing block A
IE4	No output from oxygen level sensing block B
BE1	Power supply failure
BE2	Internal failure of oxygen generator
BE3	Internal failure of decision-making block
BE4	Battery in oxygen level sensing block A failed
BE5	Sensor in oxygen level sensing block A failed
BE6	Battery in oxygen level sensing block B failed
BE7	Sensor in oxygen level sensing block B failed

Table 2-10. Parameters for the BEs and Complex BEs

BEs	Parameters	BEs	Parameters
BE1	$F_{1,2}(t) = 1 - e^{-0.00065t}$ $F_{2,1}(t) = 1 - e^{-0.00073t}$ $F_{2,5}(t) = 1 - e^{-0.00044t}$ $F_{2,5}(t) = 1 - e^{-0.00044t}$ $F_{3,2}(t) = 1 - e^{-0.00075t}$ $F_{3,5}(t) = 1 - e^{-0.00044t}$ $F_{1,4}(t) = 1 - e^{-0.00860t}$ $F_{4,5}(t) = 1 - e^{-0.00088t}$	BE2	$\lambda = 0.00023$
		BE3	$\lambda = 0.00023$
		BE4 & BE6	$\alpha(t) = 1 - e^{-0.00078t}$ $\beta(t) = 1 - e^{-0.00082t}$ $D(t) = 1 - e^{-0.00064t}$ $F_{Power}(t) = 1 - e^{-0.00285t}$
		BE5	$\lambda = 0.00015$
		BE7	$\lambda = 0.00091$

Without loss of generality, we evaluate the reliability of the system of Figure 2-39 for a mission time of 500 hours. To illustrate the effectiveness of proposed framework, we have created some scenarios as shown in Table 2-11. As can be seen at time interval [0, 100] no observation has been provided for the states of the system components. As a result, analysis performed within this interval is like an offline analysis. At time interval [101, 200], it is observed that the battery system is in state D2 and at time interval [201, 500], the power system has been observed to be in state S4. Figure 2-42 shows the reliability of the battery and power systems with and without observation. The changes in reliability of these systems due to real time monitoring is clearly reflected in the figure. For instance, for battery system and the power system, the reliability declined steadily until 100 hours and 200 hours, respectively. After 100 hours and 200 hours, respective reliability for each system drops sharply and then continues to decline steadily again. That means because of our real time observation of the battery and power system states, our

knowledge about the reliability of these systems is updated accordingly, which is not possible with design time analysis.

Table 2-11. Experimental settings

Mission Time	**Real-Time Observation**
$t = [0, 100]$	No Observation
$t = [101, 200]$	State D2 in the SMP of Battery has been observed
$t = [201, 500]$	State S4 in the SMP of Power System has been observed

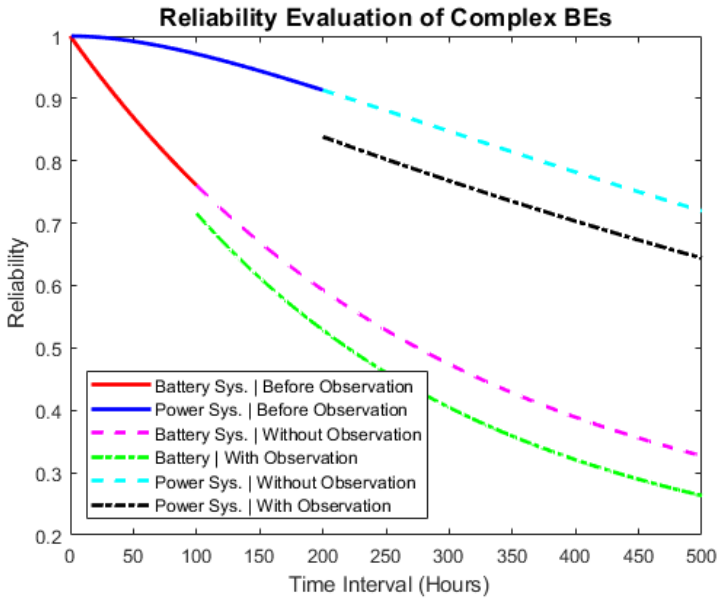


Figure 2-42. Reliability of Battery and Power systems with and without observation.

Figure 2-43 shows the reliability of the whole system for 500 hours mission time. The effects of observing the operating states of battery and power system on the reliability of the whole system is clearly visible in the figure.

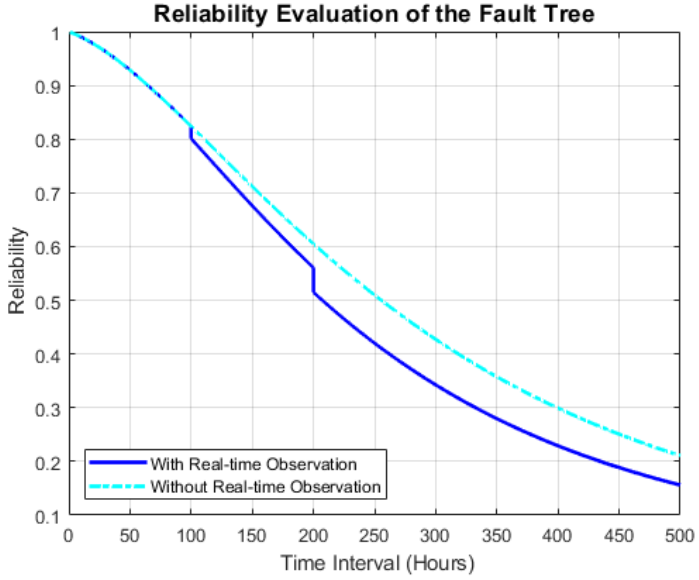


Figure 2-43. Reliability of the whole system with and without observation.

This real-time analysis feature not only helps us to update our belief about the system reliability, but also allows us to perform a meaningful analysis by taking into account the real operational status of the system.

2.12 Capabilities and Limitations of the Proposed Method

In this section, some of the capabilities and limitations, mostly related to accuracy, of the proposed method are discussed. A number of guidelines are suggested to overcome that limitation.

2.12.1 Capabilities of the Proposed Method

The proposed method offers the following capabilities:

- This thesis presents a universal semi-Markov model that can model any type of gates including static and dynamic. In addition, it embeds the functional dependencies behaviour of gates' inputs into gate model. This simplifies the final model of DFT.
- The proposed method solves DFTs hierarchically through SMP theorem. So, this method can reduce problem complexity in order to reduce state and transition explosion.
- The proposed method is able to consider non-exponential failure by means of SMP theorem. It is possible to consider hybrid failure distributions as the gate's input using the SMP theorem.
- Basic events in this method can be defined by SMP or CTMC in which repair, imperfect coverage, and other issues can be considered. This idea already exists for static FT in the literature (Kim, Ghosh, & Trivedi, 2010) and extended here for DFTs.

2.12.2 Limitations of the Proposed Method

The proposed method currently has the following limitations which could be addressed is future works.

- In the proposed method, the output of SEQ, OR, and POR gates with any complexity in their inputs will be precise. In addition, in the AND and PAND gates when a complex event or a sub-tree is connected to the first input and a simple event connected to the second input, the output will be precise. Otherwise, the output will be approximate for other gates and other situations for AND and PAND gates the output will be approximate.

- The results of this method are approximate and precision drops further for DFTs with repeated events. The proximity of the method does not necessarily diminish its effectiveness; the results obtained through this method are still more precise than those of other published research works (Yuge & Yanagi, 2008).
- In some benchmarks in which the shared or sliding spares are used, the proposed method is not able to solve shared spare gates. It is suggested to replace shared spares and use the reshaping rules (see appendix). These gates are then modelled as a semi-Markov model and generalized for any kind of failures distribution function.
- In this study, it is assumed that the fault tree has only coherent events and there would be no guarantee for a fault tree with non-coherent events. In practice, noon-coherent fault trees are avoided.
- There are still some gates like pSAND and SAND that cannot be modelled through the universal gate. We hope to improve this universal gate to consider pSAND and more other gates as future research works. It would be also possible to define new gates such as semi-PAND in the future.
- The example of repairable DFT is just provided to show that there might be a possibility to improve the approach for repairable DFTs. However, at the moment, there is no guarantee for any other example of repairable DFT. In fact, this example is just an insight into potential future works.

2.13 Conclusion

In this thesis, a novel hierarchical approach to evaluating the reliability of DFTs based on SMP theorem was presented and the universal state space model has proposed for static and dynamic gates and with inputs described by exponential and non-exponential failure distribution functions. It was shown that the proposed method competes well with other approximate solutions for reliability evaluation of DFT. A number of examples have been given to show the capabilities and limitations of the proposed solution in I) parametric solution that can be used for other related computations such as MTTF and Sensitivity, II) dealing with non-exponential failure distribution functions, III) dealing with repeated basic events, IV) no state explosion V) considering repairable events (a limited example just to provide an insight for future research works) and VI) a case study of AFDS. Moreover, the limitations and capabilities of the proposed method have been discussed.

SMP have limitations in modeling concurrency among generally distributed events. Therefore, MRGPs and phased-approximated (PH) approaches can be applied instead of SMP in the proposed method to address a wider range of applications (Distefano, Longo, & Trivedi, 2012).

Chapter 3

3 Representing Environmental Factors that in Dependability Analyses

The chapter proposes a novel model to include external environmental and accessibility variables for maintenance and availability modelling of offshore wind turbines' components. The proposal includes a method of including the estimated remaining useful life (RUL) estimations in the availability prediction models. The method is integrated with the concepts of hierarchical dynamic fault tree and complex basic events presented earlier. Application to offshore wind is being considered.

3.1 Offshore Wind Industry and its Accessibility Modelling Challenges

Offshore wind energy is a promising renewable energy source, and it is growing fast especially in Europe (Blaabjerg and Ma, 2017; Tardieu, 2017; IEA, 2019). To make the industry more successful and competitive, the Levelized Cost of Energy (LCOE) for onshore and offshore wind should be reduced. A wind turbine has four main stages in its life cycle I) Design and Development, II) Construction, III) Operation and Maintenance and IV) Decommissioning. More than 87.12 % (about 25 years) of a wind turbine's life cycle belongs to the operation and maintenance stage. As reported in (Lu *et al.*, 2009; Rademakers *et al.*, 2009; Tchakoua *et al.*, 2014), for onshore wind energy and for offshore wind energy with and without considering ageing components, the cost associated with O&M is 10-15 %, 25-30 % and up to 35%, respectively. Note that for a 1 GW wind farm, the cost of O&M including insurance and internal asset owner costs is about 75 million GBP per year (Catapult, 2019). The accessibility of a site affects maintenance and therefore a measure of this is also helpful.

Availability is a measure of the percentage of time the turbine is operational versus total lifetime (operational time + downtime) (Trivedi, Kim and Yin, 2012). Thus, unavailability gives a measure for the lost energy production because of wind turbine downtime. We note that downtime in areas with a larger resource or bigger wind farms causes significant loss of production (Martini *et al.*, 2017). For an offshore wind turbine, availability not only depends upon the failure and repair distributions of components but also depends on accessibility restrictions such as weather and logistics delays which affect maintenance. Poor accessibility can indeed make a significant difference in availability (Ge *et al.*, 2020). To illustrate the problem, consider an onshore wind turbine with availability of 97%. The same wind turbine located about 15 Km offshore and having an average inaccessibility of 25% will have about 76% of availability (Dowell *et al.*, 2013), which is a significant reduction. Moreover, in the IEC 61400-26-1 standard, the statistical availability of wind turbine has been addressed with and without accessibility consideration (*IEC 61400 - Part 26-1: Availability for wind energy generation systems*, 2019). Although accessibility can significantly influence the availability of a wind turbine, its modelling has seldom been studied in the traditional Reliability, Availability, Maintainability and Safety (RAMS) research fields.

Inappropriate weather conditions often disrupt and delay offshore operations (Diamond, 2012). Typically, such operations go ahead only under certain conditions, for example when wind speed is less than 12 meter per second and wave height is less than

2 meters (Shafiee, 2015). Furthermore, conditions like fog and poor visibility at night can make an offshore wind turbine inaccessible due to safety regulations. A study of a 500 MW offshore wind farm located in the Netherlands has revealed that 89.4% of all turbines' downtime is due to delays in repairs caused by weather (Rademakers and Braam, 2003). Because of the significant influence of weather conditions on accessibility and consequently availability, studies have considered its uncertainties and stochastic behaviours (Dinwoodie, Catterson and McMillan, 2013; Catterson *et al.*, 2016; Ioannou, Angus and Brennan, 2019; Yang *et al.*, 2020). The weather delay probability for the offshore wind operation introduced through constant values in (Byon, Ntaimo and Ding, 2010), and (Besnard, Fischer and Tjernberg, 2013) used a historical time series of weather data with two different step sizes. The seasonal effect can increase the uncertainty of the weather window probability calculation and (Santos, Teixeira and Soares, 2015) provides weather window probabilities and mean waiting time for different seasons. The wind speed distribution has been estimated through a Weibull distribution with shape equal to 1.96 and scale equal to 7.21 in (Yang *et al.*, 2020). They have also used exponential distribution for estimating the interval between opportunities and interval between delays.

Another cause of offshore wind O&M inaccessibility is the delay caused by logistics. For logistics, different factors can be considered including Crew Transfer Vessels (CTVs), service operation vessel, substation for O&M, and large component repair vessel. Smaller boats, larger ships and helicopters can be considered for crew transfer. Whilst the replacement of large components may require a ship (Catapult, 2019), ships and helicopters are expensive, and they might not be available for daily O&M. The logistics delay has been divided into two categories: on-site and off-site logistics (Besnard, Fischer and Tjernberg, 2013; Shafiee, 2015). Logistic delays have been divided into in-stock and out-of-stock logistics in (Huang *et al.*, 2017). Logistics costs can be reduced through resource sharing with cost benefits can reach up to 45% reported in (uit het Broek *et al.*, 2019) when several offshore wind farms share their resources.

Health and safety regulations impose a shift limit for O&M and it can also cause delays and specifically for large components or the components that need more than a shift to be repaired. The shift limit has been considered as a statistical constraint sub-function in the repair modelling (Besnard, Fischer and Tjernberg, 2013). This research emphasized that the shift time can be determined as 12 hours or 24 hours and did not consider night-work policies. However, working on maintenance during the night is

prohibited in offshore wind farms (Byon, Ntaimo and Ding, 2010). In (Huang *et al.*, 2017) a more advanced model is proposed which includes shift limits using a Poisson process.

The imperfect maintenance of offshore wind farms and its effect on availability has been rarely studied in the literature because of its mathematical complexity. However, imperfect maintenance models can provide a more realistic result (Alaswad and Xiang, 2017). Imperfect maintenance models can be classified into three categories. In the first category, after imperfect maintenance, the system degradation rate is increased through a random function or random variable, e.g. (Wu, Niknam and Kobza, 2015). The second category provides a number of states for the system deterioration; following imperfect maintenance, the system goes from M^{th} state to $(M-1)^{th}$ state (Ruiz-Hernández, Pinar-Pérez and Delgado-Gómez, 2020). The third category uses an analytical model in which imperfect maintenance can influence a parameter of the model like the deterioration rate of the system (Zhang, Gaudoin and Xie, 2015; Alaswad and Xiang, 2017; Kang, Sobral and Soares, 2019).

In model-based availability evaluation, the traditional approach is to use failure rate and repair rate to characterise transitions between states. However, RUL-based transitions have also been studied in (J.I. Aizpurua *et al.*, 2017; Jose Ignacio Aizpurua *et al.*, 2017) in the context of Stochastic Activity Networks (SAN). This thesis reshapes this idea into a state-space Markov model and merges it with the accessibility model proposed in (Huang *et al.*, 2017). The model is then upgraded with multi-state transition-based shift limit and imperfect maintenance to create the proposed Butterfly Maintenance Model (BMM). A MRGP is underpinning the model giving the ability to consider complex and non-exponential failure distributions. Having defined the BMM for each system's component, component models are embedded as complex basic events in a dynamic fault tree (DFT) to evaluate the whole system availability. The main contributions of this chapter can be summarised as follows:

- Proposal of a novel method that considers both traditional and data-driven availability analysis by combining both simple failure distribution and remaining useful life (RUL)-based transitions in a single MRGP. This is the first time that MRGP is combined with real-time transitions. The proposed model enables users to evaluate both condition-based maintenance and reliability-centred maintenance.
- The introduction of an integrated BMM model incorporating more advanced logistics delays, weather delays and a daily briefing and transfer delays. Consideration of these

accessibility and delay factors will lead to a more realistic availability evaluation of the offshore wind farm.

- In the Markov modelling theories, the issue of considering the time-based deadline for a transition has rarely been discussed. This chapter uses the deadline transitions to model shift time limits of the O&M for offshore wind turbines.
- Imperfect maintenance can make the model and its evaluation more realistic. However, the literature lacks proper modelling for imperfect maintenance. The proposed model is not only able to consider imperfect maintenance but also able to use this option to model the life extension of the turbine.
- The presented model can model non-exponential system behaviour using MRGP. The MRGP has higher computation complexity but at the same time offers more power in modelling complex failure behaviours. At the same time, the embedded and presented Markov model can be used with Markov processes in cases where only exponential failure and delay behaviour is considered. In other words, the computational complexity of the model is flexible and can be reduced via simplifying assumptions.
- The DFT is proposed to combine the availability models of components and evaluate the total availability of the system. Using DFTs provides an intuitive perspective on fault propagation in the system.

The rest of the chapter is organised as follows: Section 2 provides a brief literature review of the existing availability and maintenance models of a wind turbine. Section 3 defines the problem of availability modelling and evaluation studied in this chapter. Section 4 presents the proposed approach. Section 5 describes applications and numerical results highlighting capabilities and limitations. Finally, section 6 provides a conclusion for this chapter and points to future work.

3.2 Related Works

Existing approaches for the quantitative evaluation of availability, including accessibility, can be categorized into two main categories: simulation-based and model-based methods. The focus of this work is on model-based methods which generally scale up better. The review is given in chronological order and focuses on the technique/model used and the attributes of the problem covered by each method presented in a thesis.

A Markov process-based queuing model has been proposed by (Besnard, Fischer and Tjernberg, 2013) to assess the availability of the offshore wind turbine considering accessibility attributes such as the location of maintenance facilities, number and type of available CTVs, the use of helicopter and work shift limits. A Petri Nets-based parametric availability model proposed in (Santos, Teixeira and Soares, 2015) considered failure, repair, CTV logistic time, and weather delay. The article also considered Weibull distribution for rotor failure. A queuing model constructed by a multi-state Markov process has been proposed to evaluate the availability of offshore wind turbines concerning accessibility issues such as weather and transfer delays (Huang *et al.*, 2016).

The risks of installation, operation and maintenance of an offshore wind farm have been evaluated through hazard identification and Failure Modes Effects and Criticality Analysis (FMECA) in (Lazakis and Kougioumtzoglou, 2019). Important factors including safety, environmental impact, asset integrity and operation have been considered in the risk evaluation. The thesis presented a Bayesian belief Networks-based approach for availability analysis of the wind farm.

A three-state Markov chain availability model of offshore wind turbine considering accessibility problems has been proposed in (Huang *et al.*, 2017). In that model, weather delay, logistic delay and shift time limitation regarding corrective maintenance have been studied. The results of the model have also been compared with Monte Carlo simulation. It should be noted that this model was unable to consider non-exponential failure behaviour and imperfect maintenance. Moreover, the author has mentioned that the expected time for fault detection and diagnosis is too short and can be ignored. In (Catelani *et al.*, 2020), authors report cases and situations where 26% of the maintenance time was consumed by fault-finding. Paul et al. (Paul and Rather, 2018) have used an approach to select a suitable wind turbine for a site based on four attributes: economy, reliability, resilience, and environment. This study has applied a Markov chain to model the transition of damaged turbine states and adopted expected energy not supplied and availability to use in their optimization procedure.

A Markov decision process-based method has been used to model the corrective maintenance of an offshore wind farm considering different policies by (Seyr and Muskulus, 2019b). The thesis considered actions such as stay, wait, reset wait time, go out, repair and return. The proposed Markov decision process has been used as a core for the optimization of maintenance planning. Over 135 research thesiss regarding the use of

decision support models for the scheduling of maintenance in offshore wind farm have been reviewed by (Seyr and Muskulus, 2019a).

The thesis summarized the Key Performance Indices (KPIs) and parameters influential to the offshore wind farm operation and maintenance scheduling. In addition, various degradation and failure models in this field have been addressed. The article categorized these models into three classes intended for Preventive Maintenance, Condition-based Maintenance, and Corrective Maintenance. Furthermore, the models have been classified into Monte Carlo simulation, Discrete Event models, Markov Models, Petri nets, and Poisson process. Ruiz-Hernández et al. (Ruiz-Hernández, Pinar-Pérez and Delgado-Gómez, 2020) have focused on O&M cost reduction and proposed maintenance sequential tasks allocation. The thesis modelled the imperfect and preventive maintenance using graph theory. However, the used model can also be considered as a multi-state Markov process.

Fault Tree Analysis is one of the well-known approaches for reliability and availability evaluation because of its simple and powerful illustration of fault propagation. A Fault Tree (FT) can visualize failure combinations of the system in a top-down deductive way. (García Márquez *et al.*, 2012; Artigao *et al.*, 2018; Kang, Sun and Guedes Soares, 2019; Li and Coolen, 2019; Adumene and Okoro, 2020; Konstantinidis, Katsavounis and Botsaris, 2020) have provided fault tree models for offshore wind turbines.

Dynamic Fault Tree (DFT) is an extension of Fault Trees which can model dynamic failure behaviours such as priority, spares and sequence (Dugan, Bavuso and Boyd, 1992). A DFT for offshore wind turbines is provided in (Asghari, Mohammad and Oskouyi, 2015; Zhang *et al.*, 2016) and (Sun *et al.*, 2023) combined DFT with working condition correlations. However, although FTA-based models are used for reliability and availability evaluation of offshore wind farms, maintenance and accessibility are ignored.

Moreover, in FTA, generally, the basic events of fault tree are characterised by failure rates or failure probabilities and repair rates. As a result of this, the complex failure behaviour of system components and other associated issues are usually ignored.

To overcome this limitation, in this thesis we use DFTs that model dynamics and in which complex behaviours of system components are modelled as complex MRGP-based events. (Leimeister and Kolios, 2018) have reviewed existing models for risk and reliability evaluation of offshore wind turbines.

(Abeynayake *et al.*, 2021) has proposed a Markov-based solution for availability evaluation of large-scale offshore wind farms and included their collector system. Although the approach was suitable for large-scale systems but it has its own limit for considering the different maintenance and accessibility parameters. (Zhu *et al.*, 2023) has focused on long-term reliability and availability evaluation of wind turbines and used a Markov-based approach for quantitative evaluation. However, the author did not consider the accessibility issues.

Regarding OWTs, two types of availability are generally considered; I) power availability where the power curve is studied using metrological inputs (Göçmen et al., 2019; Vahidzadeh and Markfort, 2019; Wang et al., 2019; Nielson et al., 2020) and II) operational availability where the effect of failure and repair plus maintenance strategies are addressed. This thesis deals with the second type that should not be confused with the first type.

Table 3-1 provides a mapping of the key model-based methods for availability evaluation and the key aspects they address.

Table 3-1. A Comparison of Model-based Availability Evaluation Approaches

	(Besnard, F. et. Al.)	(Huang, L. L. et. Al.)	(Lazakis, I. et al.)	(Ossai, C.I. et al.)	(Chen. Z. et al.)	(Santos, F. et al.)	(Le, B. et al.)	(Seyr, H. et. al.)
Proposed	(Besnard, Fischer and Tjernberg, 2013)	(Huang *et al.*, 2017)	(Lazakis and Kougioumtzoglou, 2019)	(Ossai, Boswell and Davies, 2016)	(Chen *et al.*, 2019)	(Santos, Teixeira and Soares, 2015)	(Le and Andrews, 2016)	(Seyr and Muskulus, 2019b)

Method ► Aspects ▼	Multi-State Markov Chain	Markov Chain	Bayesian Network	Markov Chain	Hidden Markov Model	Petri Nets	Petri Nets	Markov Decision Process	DFT Semi-Markov Process +
Logistics Delay	✓	✓				✓	✓	✓	✓
Weather Delay	✓	✓				✓	✓	✓	✓
Transfer Time	✓	✓				✓	✓	✓	✓
Repair Time	✓	✓	✓	✓	✓	✓	✓	✓	✓
Shift Time Consideration		✓							✓
Importance Measure							✓	✓	✓
Imperfect Maintenance				✓					✓
Real-time RUL					✓				✓
Costs			✓		✓		✓	✓	✓
Accessibility	✓	✓					✓		✓
Non-Exponential Distribution						✓			✓

The detailed analysis of the state-of-the-art shows that no single model has yet considered the following issues: I) non-exponential distribution consideration through MRGP, II) both traditional and data-driven Markov transitions, III) modelling of shift limits and IV) imperfect maintenance.

The integrated proposed method develops the comprehensive BMM model to capture and evaluate the availability of components in OWTs. Using this rich model as a complex basic event of a DFT enables the users to capture and evaluate accurately the availability of the turbine under complex assumptions of maintenance.

3.3 Problem Definition

The problem definition is divided into three parts: availability and related KPIs, condition monitoring and estimation of RUL, and impact of accessibility factors.

3.3.1 Availability and related KPIs

Time-based steady-state availability can be defined as (3-1) considering the operational time and total lifetime of a turbine. The best-case scenario for a wind turbine is to be always operational and available.

$$A(\infty) = \frac{Time\ Available}{Total\ Time\ Considered} = \frac{Operational\ Time}{Operational\ time\ +\ Downtime} \quad (3\text{-}1)$$

The steady-state production-based availability (PBA) can be defined as in (3-2), i.e. as the fraction of actual generated energy over the potentially expected energy production including energy lost when the turbine was not operational due to inspection, failure and maintenance.

$$P = \frac{Actual\ Energy\ Produced}{Potentially\ Expected\ Energy\ Production} \tag{3-2}$$

Although our modelling focus is on availability, we also provide in (3-3) a simple calculation of the Cost (C) of unavailability in power production, for the purposes of studies of different models of maintenance on availability and thus cost.

$$C = C_f * \left(1 - A(\infty)\right) \times OWFCapacity \times Days \times Hours \tag{3-3}$$

Where C_f is the cost of MW/h electricity production, and *OWFCapacity* is the total capacity of the wind turbine or wind farm. It is also possible to calculate the effect of carbon emission reduction based on availability which is not considered in this thesis.

3.3.2 Condition Monitoring and Fault Prognostic

Accessibility difficulties mean that data collection and condition monitoring have a vital role in offshore wind. Supervisory Control and Data Acquisition (SCADA) and Condition Monitoring Systems (CMS) are widely used in this industry. SCADA typically generates time series that includes sensor measurements such as temperature, pressure, vibration, current, rotational speed, while CMS typically further processes such data. For instance, Envelopes, FFTs (Fast Fourier Transforms), and RMS (Root Mean Square) of the vibration signal can be the output of the CMS system. Generally, Condition monitoring can be divided into four steps; I) Data acquisition, II) pre-processing, III) feature extraction and IV) feature reduction. In addition, fault detection can be presented in three steps; I) Fault Detection, II) Fault Isolation and III) Fault Identification. Based on the result of fault identification, the nature of fault can be categorized as abrupt, incipient and intermittent. It should be noted that fault prognostics is usually applied on incipient faults. For incipient faults, remaining useful life can be predicted and used for preventive maintenance.

On the other hand, when the nature of the fault is abrupt corrective maintenance in the form of repair is inevitable (Sikorska, Hodkiewicz and Ma, 2011). Figure 3-1 shows a general scheme of a system's health degradation. This figure is divided into three zones; (I) Healthy Zone in which system has the highest operational performance. (II) Degradation Zone starting from an initiation event. The loss of performance is acceptable in this range and usually, early condition monitoring and detection can be performed in

this zone; Detection through (a) vibration analysis, (b) oil analysis, (c) audible noise, (d) abnormal pressure, (e) abnormal temperature. (III) Failure Zone in which components or a system may experience extreme loss of performance and symptoms like smoke and crack. Depending on the functionality and behaviour of the system, a region can be selected for the low-cost preventive maintenance with moderate and acceptable risk.

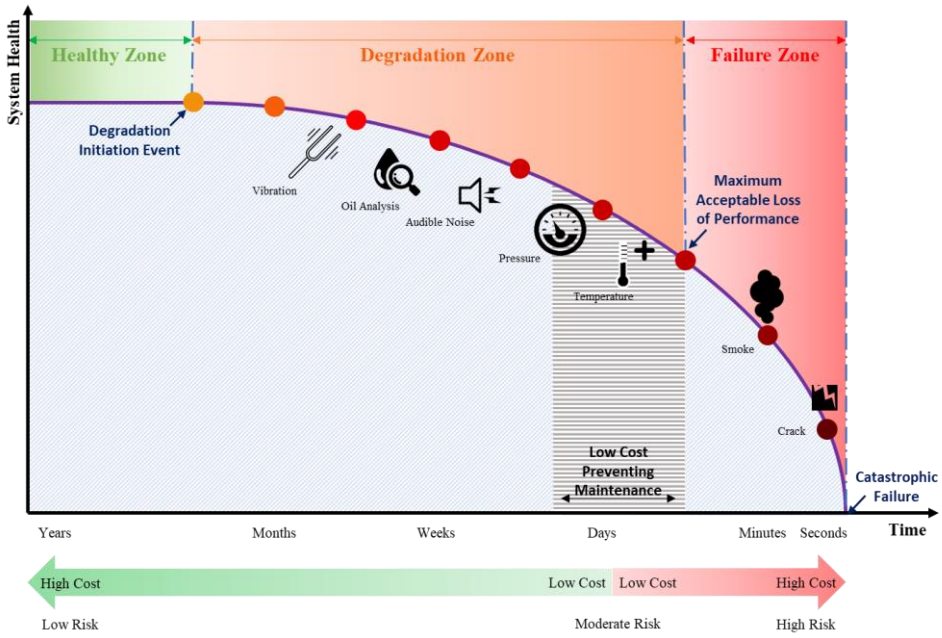


Figure 3-1.Zones in System Health Degradation

Using condition monitoring and estimation of system health degradation, it is possible to provide condition-based maintenance. Aizpurua et al. have studied a possible way of considering estimated RUL and generating a RUL-based failure distribution function (J.I. Aizpurua *et al.*, 2017; Jose Ignacio Aizpurua *et al.*, 2017). The RUL-based failure distribution function G(t) can be presented as (3-4).

$$G(t) = 1 - e^{-\left(\frac{t}{RUL(T_P) - SF - PD}\right)} \quad (3\text{-}4)$$

Where SF is the safety factor which is introduced to model the uncertainty of the RUL estimation and the time needed to trigger the maintenance. PD is the predicted delay introduced to model the effect of accessibility issues causes of delay. For example, if the RUL for a component is 20 days. one may consider one day as safety factor. The PD factor can be determined through average or expected delay time associated with accessibility issues over a period of time. Both SF and PD factors can be adaptively tuned during the maintenance procedure. The accessibility issues will be discussed in the next section.

3.3.3 *Maintenance Procedure and accessibility*

A typical offshore wind farm workflow has been described in (Koltsidopoulos Papatzimos, Dawood and Thies, 2017; Koltsidopoulos Papatzimos, 2020) and the key stages are shown in Figure 3-2.

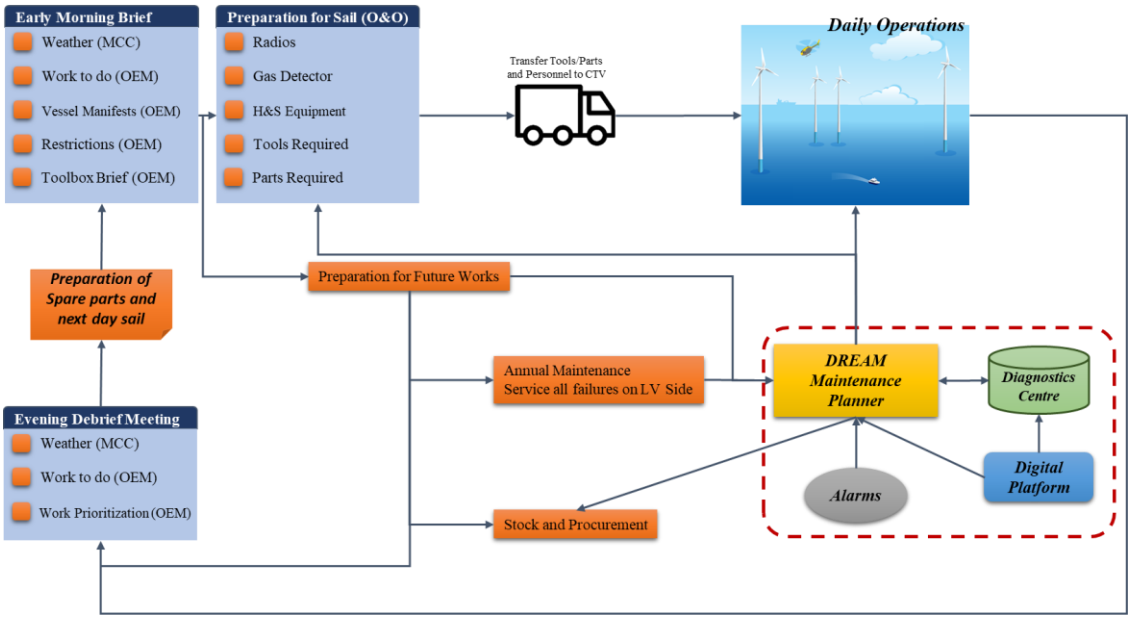


Figure 3-2. A Typical Operation and Management Procedure of an Offshore Wind Farm (Aslansefat and Papadopoulos, 2020)

An offshore operation and maintenance team start their work with early morning brief in which the weather status will be addressed by Marine Control Coordinator (MCC). A list of tasks, vessel manifest, restrictions and toolbox brief is provided by Original Equipment Manufacturer (OEM). The next step is preparation for sail if the weather permits. Owner and Operator (O&O) will determine the requirement for a sail. In addition, the maintenance plans are determined by the Offshore Control Coordinator using information from the diagnostics centre including alarms raised. At the end of the day, an evening debrief meeting will be held in which next day's weather, work, and prioritization of tasks will be addressed.

3.3.3.1 *Delay for Vessel and Crew Availability*

Depending on the type of failure, different experts and vessels may be required. As an example, the failure of the turbine blade needs experts on replacement or repair. It also requires a bigger vessel with a crane to replace the failed blade. In contrast, failure in the gas detector sensor can be handled with a normal crew and any available vessel. There are some cases in which an expert crew or appropriate vessel are not available on-demand. Thus, this kind of unavailability can cause delay and consequently reduce the availability

of the turbine (Chiachío-Ruano, Hermile and Kolios, 2019; Kolios, 2019; Kolios *et al.*, 2019).

3.3.3.2 Transfer Time

Transfer time is usually consisting of transporting the maintenance group to the port and from the port to the wind farm. The transfer delay is contingent upon the type of vessel. In some cases, the maintenance crew can be transferred to the wind farm through helicopter which is faster. As the variance of transfer time is low, it is better to use the uniform distribution function instead of exponential distribution function for delay consideration. New generation offshore wind turbines with a higher rate of capacity and power generation can increase the delay of transfer time because they need to be installed in a far location which has more wind power potential. According to a System Performance, Availability and Reliability Trend Analysis (SPARTA) benchmarking system report, in the UK, less than 25% of installed offshore turbines have less than 10km distance to shore, about 50% of offshore turbines have 10-30 km distance to shore and more than 25% of offshore wind turbines have more than 30 km distance to shore (ORE Catapult UK, 2019).

3.3.3.3 Logistics Delay

The accommodation of maintenance crews can be on-site or off-site (Shafiee, 2015). The distance of this from points where maintenance is required introduces a delay. In addition, spare parts that are out of stock may introduce further logistics delay (Huang *et al.*, 2017). In this thesis, we consider on-site, off-site and out of stock delays.

3.3.3.4 Weather Delay

When wind speed is higher than 12 meter per second or wave height is higher than 2 meters, the wind turbine will be inaccessible [14]. This inaccessibility can cause delay and consequently reduce the availability of the OWT. An example of delay estimation using the interval between opportunities and intervals between delays can be found in (Yang *et al.*, 2020). Regarding the RUL-based predictive maintenance, it is necessary to use weather forecast and plan the maintenance (Yürüşen *et al.*, 2020). Weather delay can differ based on seasons and as expected in winter there are higher (Santos, Teixeira and Soares, 2015). Online platforms such as (*BMT ARGOSS - WaveClimate*, no date) thcan provide useful weather forecast and statistics. Thus, state of the sea and consequent weather delays can be estimated using the forecasts given on these platforms.

3.3.3.5 Repair Time

Depending on the severity of the fault, a component may need a minor repair, major repair or even replacement. As expected, minor repair has a smaller repair rate than a major repair. It is also clear that replacement of large components such as blade, rotor, generator, and gearbox may cause significant unavailability. However, the replacement of small components like a sensor or a control board should be quick. In this regard, references like (Carroll, McDonald and McMillan, 2016a) and (Faulstich, Hahn and Tavner, 2011a) have provided failure rates, repair rates and costs for different components of an OWF. It should be noted the statistical uncertainty in the estimation of those parameters for components can affect the estimated availability of the system (Scheu *et al.*, 2017).

3.3.3.6 Imperfect Maintenance

With corrective maintenance (either minor or major repair) the system can be operational again. However, there are some cases that corrective maintenance does not fully restore the original state. In addition, a failed component is often replaced by a second-handed component originating from the decommissioning of another turbine. This thesis uses a factor called Betta denoting the imperfect coverage probability and the secondary failure distribution that represents the failure behaviour of the component after imperfect maintenance. The secondary failure distribution function enables users to use BMM also for availability evaluation of different life-extension strategies such as repowering, remanufacturing, and refurbishment (Shafiee and Animah, 2017; Ziegler *et al.*, 2018).

3.4 Butterfly Maintenance Model (BMM) Approach

This section presents a novel approach for availability evaluation of offshore wind turbines considering complex behaviours and accessibility issues. The BMM approach has four main sections. Figure 3 illustrates the flowchart of the BMM method. The first section is the fault tree generation using a well-known model-based safety analysis tool called Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS)(Papadopoulos *et al.*, 2011, 2016; Kabir, Aslansefat, *et al.*, 2019; Mian *et al.*, 2019). At the beginning of the procedure, a proper fault tree model for offshore wind turbine should be selected. Having a proper fault tree model for offshore wind turbine requires significant time and many expert meetings. In addition, given the rapid growth of the wind industry, it would be challenging to provide a generically applicable fault tree

model of OWT. In this case, the fault tree model can be generated using HiP-HOPS. In this tool, the user needs to provide model of the architecture of the OWT (e.g. in SIMULINK) annotated with the failure behaviour of the components. Then HiP-HOPS can automatically generate the fault tree and failure mode and effect analysis (FMEA) of OWT.

Having generated the fault tree, the second step involves processing data from the prognostics unit of a component and estimating the remaining useful life (RUL) of that component. The estimated RUL-based probability distribution is used as a transition in the Butterfly maintenance model of the component described later here. It should be noted that in the case that RUL is not available, RUL value should be set to zero and the model will automatically adapt itself to consider traditional reliability parameters only.

Accessibility issues are considered in a third step. The availability of spare parts for each basic event of the fault tree is obtained from asset management and logistics software. A range of factors that limit accessibility limits are optionally considered in the Butterfly Maintenance Model. For example, delay for vessel availability, delay for expert crew availability, delay for a shift or working hour limits, delay for logistics and preparing spare parts, and delay to find an appropriate weather window.

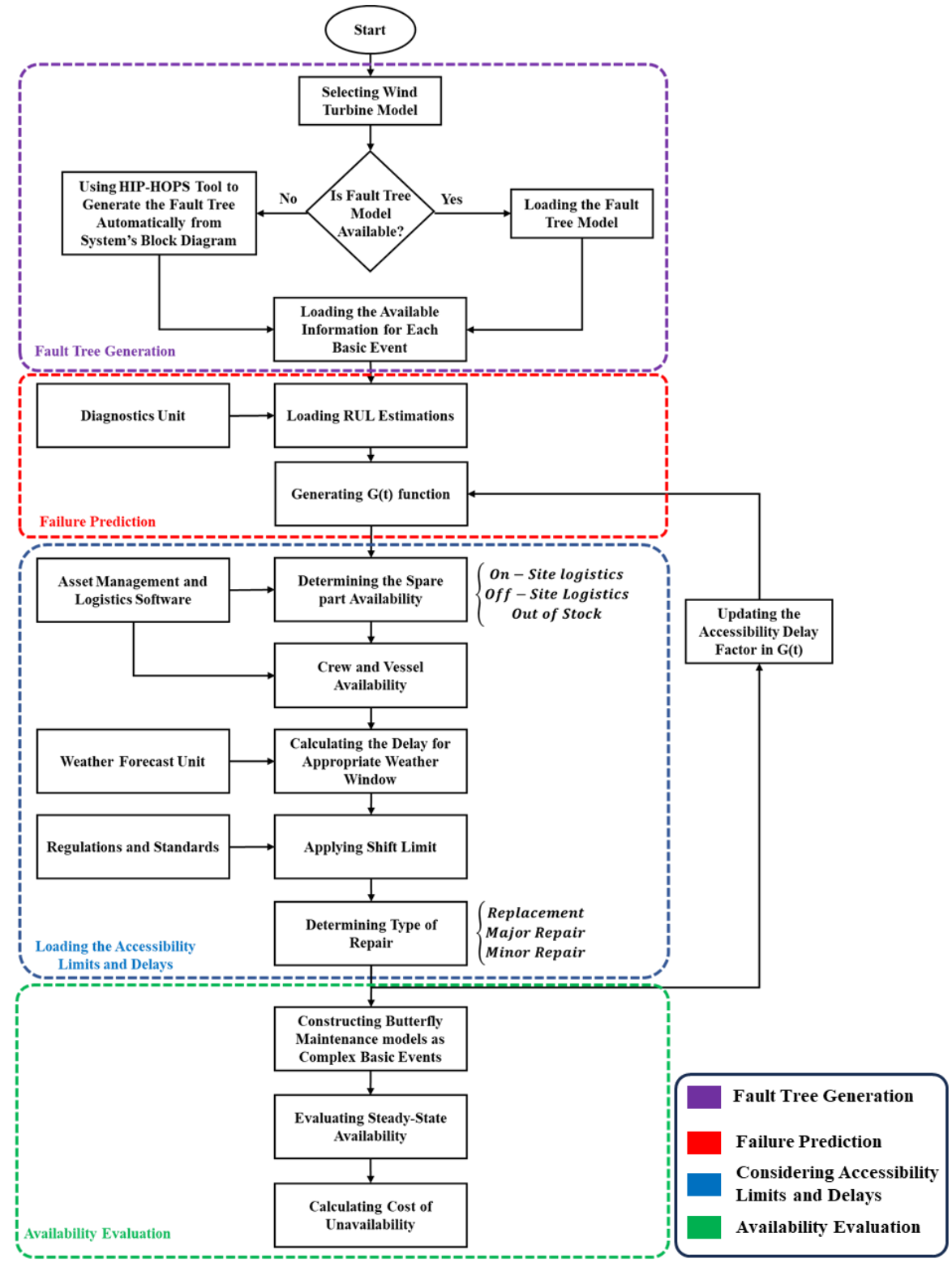


Figure 3-3. Flowchart of the BMM method

Delays in calling expert crew, finding appropriate vessels, and availability of spare parts can be fetched from asset management and logistics software. For example, if we consider a blade failure, then such software may reveal that spare blade parts are available on off-site logistics which takes one day to prepare for replacement. For replacement, a

large vessel with a crane would be needed that is available after two days. Also, a team of expert crews are needed to install a new blade and they will be available in one day.

Moreover, based on historical data, the needed time for blade replacement can be projected, and if it is more than a shift time of 8, 10, or 12 hours, then delay of the shift limits should be added. As shown in the flowchart, a set of regulations including shift limits and night work prohibition can be set in the framework to calculate the actual accessibility delay more accurately. Using weather forecast, the appropriate weather window and its corresponding delay will also be determined. Considering all delay factors, the PD and SF values of RUL function (in section 3.3.2) can be adaptively tuned to cover these delays and improve the availability of the OWT. For example, consider based on RUL estimation, a component may fail after 30 days. So, based on criticality of the component, SF can be set between 1 to 5 days. The PD factor can also be considered using expected value of all possible accessibility delays.

In the next step of the process, Butterfly Maintenance Models are populated with parameters. These are used as complex basic events in the fault tree. The maintenance model is created for each part of the system individually. There are some cases in which the required data is not available for basic events and instead data is available for intermediate events. In such cases, the Butterfly Maintenance Model should be created for intermediate event considering the equivalent failure rate for that intermediate event. An example of this is included in this thesis. The steady-state availability of each Butterfly Maintenance Model is then calculated and using the fault tree, the unavailability of the whole system can be achieved. Based on the system's unavailability the cost and other related parameters can be extracted.

3.4.1 Complex Event Construction as BMM using RGMP

Before creating the model, it is necessary to clarify the assumptions. Therefore, in the following model construction, it is assumed that:

- The initial states of the system are operational, i.e. the system is healthy. This assumption helps determine the initial probability vector of any Markov model.
- There are no dependencies between complex basic events (CBEs) and therefore their corresponding BMMs. Having a correlation or interconnection between different BMMs reduces the accuracy of the availability evaluation. For example, consider a repair team fixing one component, if in the middle of the work they start to repair

another component then the model will not be valid. In other words, the repair operation of the OWT should be sequential.

- Common cause failures and their related issues are not considered in the BMM model.
- RUL source signals should be independent and identically distributed (IID). Consider having two RULs for two different components, if there is a correlation between source siganls of each RUL, then RULs might also have correlation. In this situation a coherent fault tree cannot be used. The non-IID signal should be considered in a single model with Multivariate inputs.
- Optimal threshold design and its performance assessment are not considered in this study; for further information th reader is referred to (Aslansefat *et al.*, 2020).

A basic, three-state Markov chain availability model of offshore wind turbine has been presented by (Huang, Fu, Mi, Cao, & Wang, 2017). In this thesis, the model upgraded to an MRGP to be able to consider wider range of failure, delay and repair distributions as shown in Figure 3-4. In this model "Op", "F", and "M" stands for operational, failed and ready for maintenance respectively. Moreover, $\mathrm{F}(t), \alpha(t)$ and $\mathrm{R}(t)$ denote failure, delay and repair distribution functions respectively.

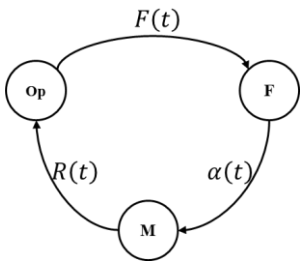


Figure 3-4. The availability MRGP hat considers failure, delay and repair with non-exponential distribution.

The idea of using both RUL-based and traditional failure rates in one model has been proposed in (J.I. Aizpurua *et al.*, 2017; Jose Ignacio Aizpurua *et al.*, 2017). The original idea has used the SAN (Stochastic Activity Networks) model but for the purposes of our work, we have developed the MRGP version of it as show in Figure 5. In this model G, F, R, and R' are representing RUL-based failure Cumulative Distribution Function (CDF), traditional failure CDF, repair CDF for corrective maintenance and repair CDF for predictive maintenance respectively.

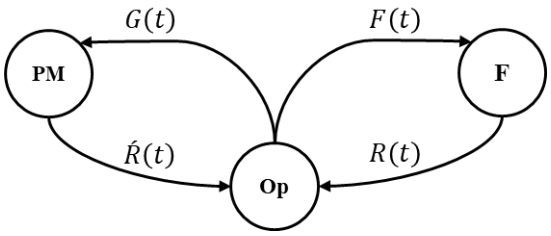


Figure 3-5. The availability model that considers both traditional failure rate and RUL-based transition

It is now possible to merge the two models of Figures 4 and 5 to produce the more nuanced model of Figure 6. All models are represented as MRGPs for considering more realistic non-exponential transitions. In the combined model, the alpha distribution represents all delays including weather delay, logistics delay, crew transfer, etc. It should be noted that the repair CDFs can be similar for some scenarios and can be different for some other scenarios. In this thesis we assumed that the repair CDFs from both paths are identical. In general, the model is capable of considering different repair CDFs.

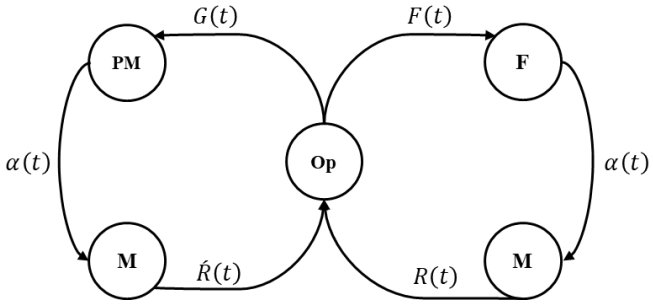


Figure 3-6. The primary BMM availability model

If the crew transfer flow duration follows a uniform distribution function, then it can be considered separately as illustrated in Figure 3-7.

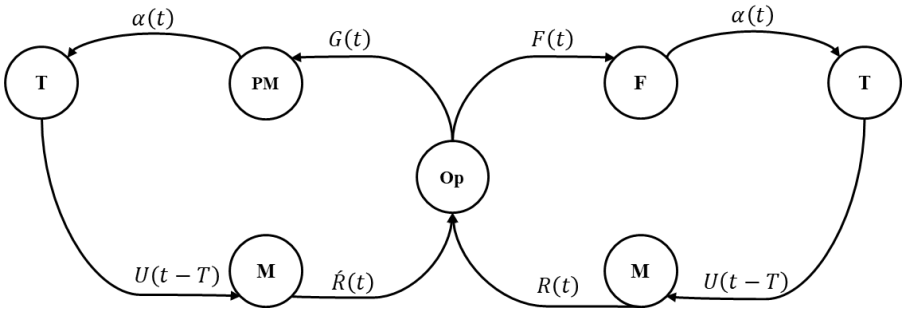


Figure 3-7. The BMM availability model considering separated uniform delays (crew transfer delay)

The shift limit (see section 1.3) is also considered to make the model realistic. Figure 3-8 shows the MRGP with n shift-states. In state “M1” if the repair finishes before deadline d, the system will go to the operational state, but if the repair takes more than d the system goes to a new state “M2”. This procedure continues to create new states until the repair finishes before the deadline.

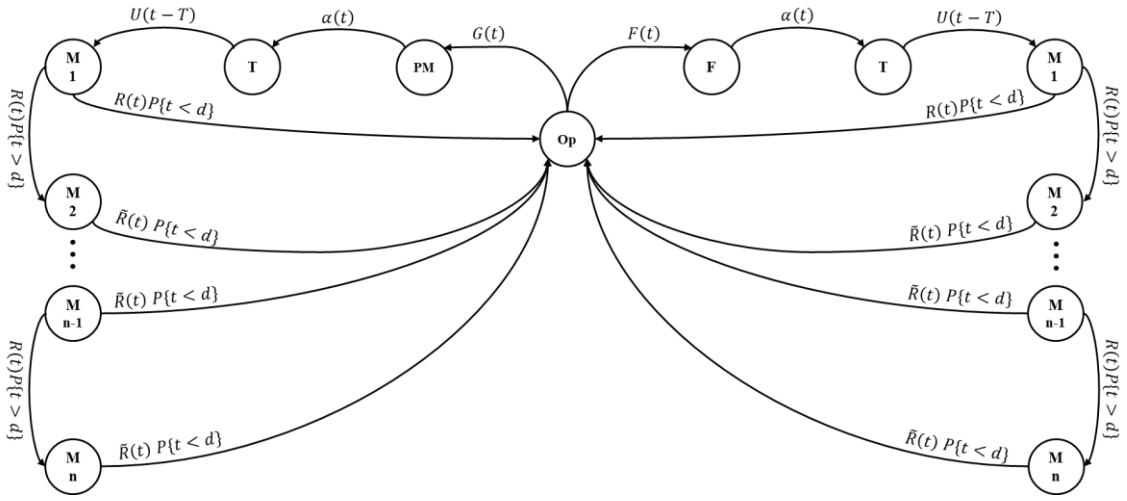


Figure 3-8. The BMM availability model considering the shift time limitation.

Figure 3-9 shows the fully developed availability model that includes an imperfect maintenance factor β together with all other factors mentioned earlier. In this model, state Op1 is a degraded operational state following imperfect maintenance in which the system is not working as well as in "Op" state and it may fail sooner.

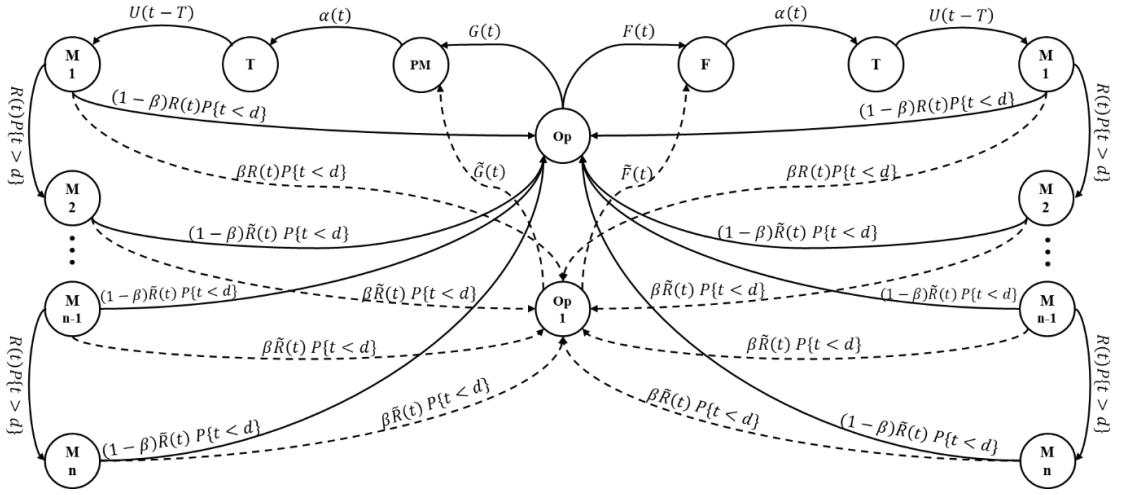


Figure 3-9. Availability model (Butterfly Maintenance Model) for a component with imperfect maintenance (Aslansefat and Papadopoulos, 2019).

For the purposes of availability modelling for the OWT, the availability model of each component must be created. Assuming these models are independent, then each model becomes a CBE in the Dynamic Fault Tree of the OWT. For more information regarding Fault Trees with CBEs please refer to (Kabir, Aslansefat, *et al.*, 2019). It would be unrealistic to assume that RUL-estimation will exist for all components. Thus, the left side of the BMM will be removed when the G(t) is equal to zero or there is no RUL-estimation available as this is shown in Figure 3-10.

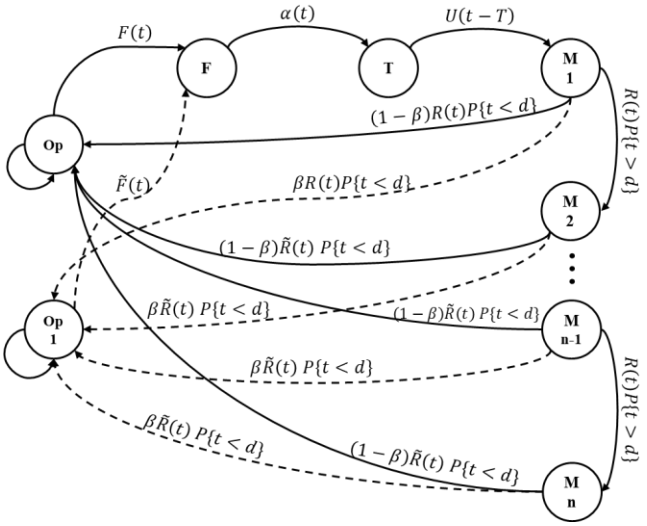


Figure 3-10. Availability model (Butterfly Maintenance Model) when the RUL-estimation is not available.

3.4.2 Incorporating Complex events in FTs

Once a BMM has been created for each component of the WT, the dynamic fault tree is then used to provide the availability of the WT as shown in Figure 3-11.

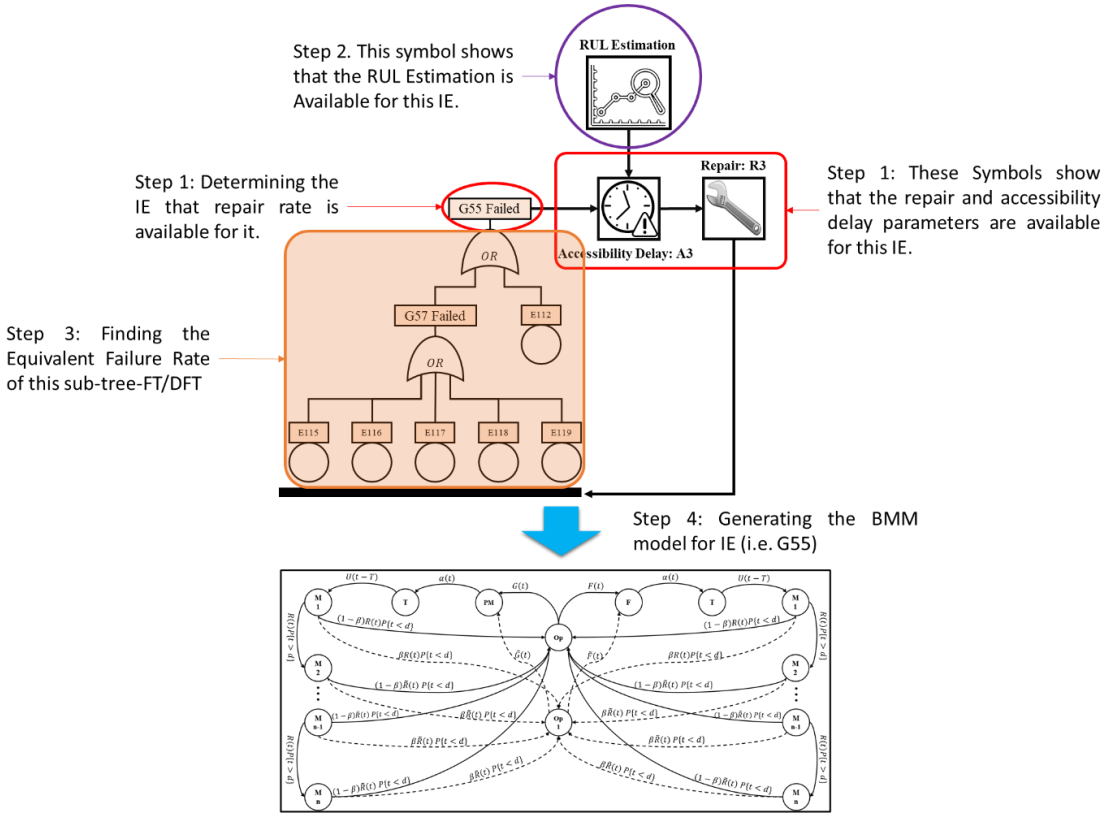


Figure 3-11. Stepwise procedure of generating BMM

Note that the advantage of the use of DFT over classical fault tree is that it allows to model complex failure behaviour of systems by considering issues such as dependencies and effects of ordering between failure events and modelling different types of redundancy profiles. In a DFT, the top of the tree (TE) represents a system failure

condition. In this thesis, this failure condition can be the failure of a whole wind turbine system or a subsystem within the wind turbine system. The TE is decomposed into a set of intermediate events (IEs), which are the immediate causes of the TE. These IEs are further decomposed down to lower-level IEs and at lowest-level to basic events (BEs). Usually, BEs represent the failure of system components. To evaluate the availability of the wind turbine, we need to evaluate the probability of the TE of the FT and to do this we need to provide quantitative data for BEs.

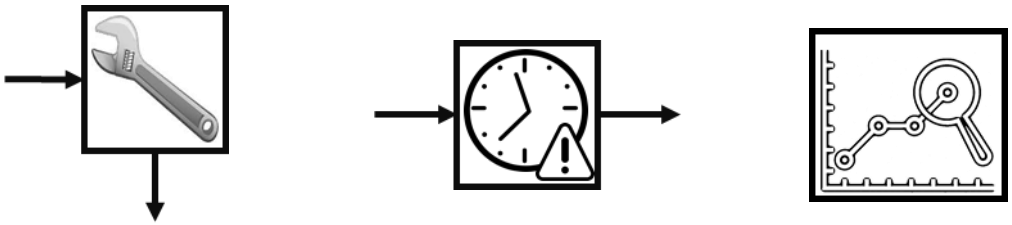
Unlike in other areas, in the offshore wind industry, we need to consider complex failure, repair and maintenance procedures to define the failure behaviour of a component. And for this we have proposed the complex BMM model defined in section 3.4.1 that represents the behaviour of a component either as BE or as IE. After that, the DFT can be solved hierarchically to obtain the unavailability of the system. For more detail regarding the hierarchical solution of DFT, see (Aslansefat and Latif-Shabgahi, 2019b; Kabir, Aslansefat, *et al.*, 2019).

There can be two scenarios for incorporating complex events in DFT; I) scenario 1 is when quantitative information is not available for all BEs and is there is sufficient information for IEs (see Example 2) and II) scenario 2 is when all needed information is available for BEs (see Example 3).

Scenario 1: Typically, for wind turbines failure rates are available for BEs. However, in most of cases, maintenance data and RUL estimations are not available for BE level. In this situation, the following steps should be taken to be able to incorporate BMMs in the DFT of the system.

1. Determining the IEs that repair rates are available for them.
2. Checking the availability of RUL estimation for those selected IEs in previous step.
3. A sub-FT bellow each selected should be removed and its equivalent failure rate should be stored.
4. A BMM model for each selected IE should be created based on: event repair rate (step 1), RUL (step 2: if RUL estimation is not available set G(t) to zero), equivalent failure rate (step 3), imperfect maintenance (if beta is not available then set it to one), and accessibility delay parameters. It should be mentioned that the accessibility delay parameters such as weather delay, and transfer delay are the same for whole fault tree.
5. Solving the FT/DFT to obtain the availability of OWT using the theory explained in the previous section.

To decrease the complexity of the procdure, three symbols have been invented to improve the understading of the FT model in case of having scenario 1. It should be mentioned that for scenario 2, the traditional DFT model and its symbols will be used. Figure 3-12 (a) illustrates the schematic of a repair gate in FT. In this gate, the input comes from an intermediate event in FT and the output goes to a basic event or a set of basic events. For each repair gate, the repair rate, imperfect maintenance coefficient and its probability distribution function should be defined. This gate can simply show the level of repair information availability in a FT. Regarding the accessibility consideration, a delay block or gate is proposed to model different delays. For this gate, the aforementioned delays for offshore wind and their probability distribution function can be defined. The schematic of this gate is shown in Figure 3-12 (b). Finally, Figure 12(c) shows a symbol of declaring that RUL estimation is available for the event on the fault tree connected to it.



(a) (b) (c)

Figure 3-12. Schematic of (a) Repair Gate (b) Accessibility Delay Block in and (c) RUL availability in FT

Scenario 2: In this scenario, the required information for generating a BMM model is available for all BEs. In this case there is no need to convert any part of FT/DFT and we can directly incorporate the BMMs as complex BEs. As mentioned before, if the RUL estimation is not available for an BE, G(t) in the associated BMM should be set to zero.

3.5 Results

The capabilities and numerical analysis of the proposed model are addressed in this section. Example one is addressing the capabilities of the butterfly maintenance model for availability evaluation of a 2MW wind turbine considering only the failure of the high-speed bearing. The second example demonstrates the combination of butterfly maintenance model and dynamic fault tree for evaluating the availability of the system.

3.5.1 Example 1. High-speed Bearing in 2MW Wind Turbine

In this example, we apply the Butterfly Maintenance Model (BMM) for the availability evaluation of a high-speed bearing component of a WT. A real-time exponential degradation-based model has been used to predict RUL (*Wind Turbine High-*

Speed Bearing Prognosis - MATLAB & Simulink - MathWorks United Kingdom, no date). An open-source dataset of a high-speed shaft in 2MW wind turbine from (Bechhoefer, Van Hecke and He, 2013) has been used and from this dataset, for 50 consecutive days, a 6-second vibration signal is considered. It is known that an inner race fault developed and caused the bearing failure on day 50. Using the exponential degradation model, the fitted health indicator curve is obtained. Having fitted the health indicator, a threshold should be defined before calculating the RUL (as explained in section 3.3.2). In this example, Train-test breakpoint has been set to 20 days, and after 20 days the model starts to predict, and day by day adapt (i.e. correct itself) to predict more accurately. Figure 3-13 shows both true RUL and estimated RUL with their α bound 20% as Confidence Interval (CI).

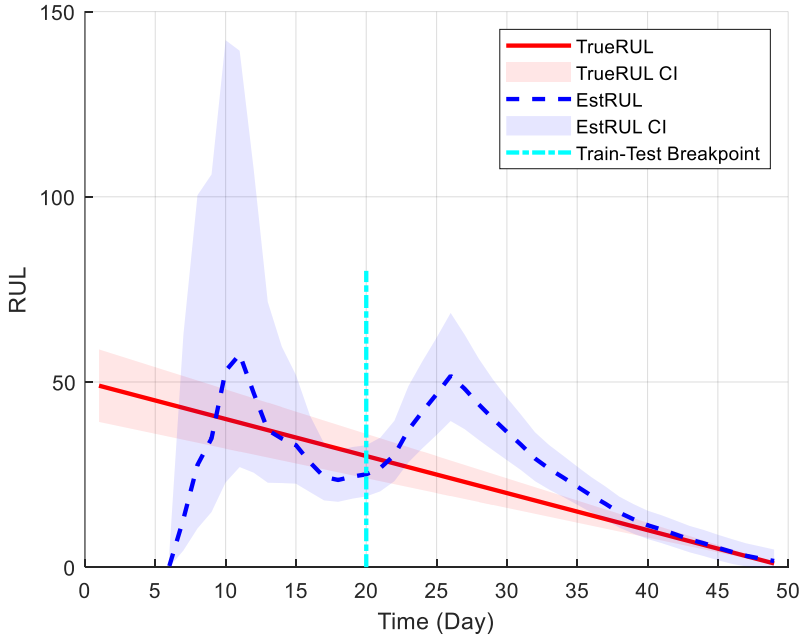


Figure 3-13. True and estimated RUL with α bound 20 percent.

Table 3-2 provides the parameters used in the BMM. For example, the delay rate is 1e-2 which includes all accessibility delays as 100 hours with exponential distribution function. Repair rate for both RUL-based side of BMM and traditional side is the same as 1e-3. Thus, the meantime to repair is 1000 hours in this case. It is assumed that the repair CDF is equal for both sides of the model.

Table 3-2. Parameters values and distribution type in BMM (Example 1)

Parameter	Distribution Type	Rate
Failure F(t)	Exponential	5.20e-6
RUL-based Failure G(t)	Exponential	Using Figure 3-13 and Equation 3-4
Repair CDF R(t)	Exponential	1e-3
Delay CDF	Exponential	1e-2

Transfer Delay	Uniform	2 Hours

Using the BMM model of Figure 3-7 and the parameters of Table 3-2, the availability of wind turbine is plotted in Figure 3-14, considering only a bearing fault, perfect maintenance and no limitation in shift time. The horizontal axis denotes the remaining time to failure (days) and the vertical axis provides the system availability. Note that the component (&WT) fail when RUL is zero.

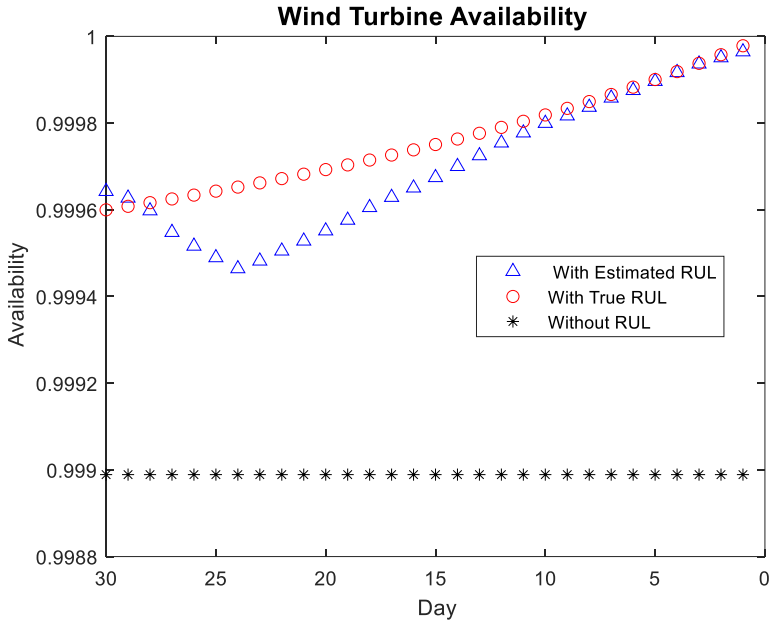


Figure 3-14. Wind Turbine Availability vs. remaining useful life considering only bearing faults, perfect maintenance and no limitation in shift times.

It should be noted that the accuracy of RUL estimation can affect the availability of WT. For instance, with low accuracy RUL, the condition-based maintenance can start early or late that not only affect availability but also can increase the risk of failure or the cost of maintenance as shown in Figure 3-1. It is also clear that without considering RUL estimation and using only reliability-centred maintenance the system will gain less steady-state availability. It is important to notice when for any reason RUL signal disappears, the reliability-centred part can still handle the system's maintenance, albeit with less availability.

Considering 176 GBP as a cost for energy loss per MW per day, the annual cost of energy loss in a WT using steady-state unavailability can be illustrated as Figure 3-15.

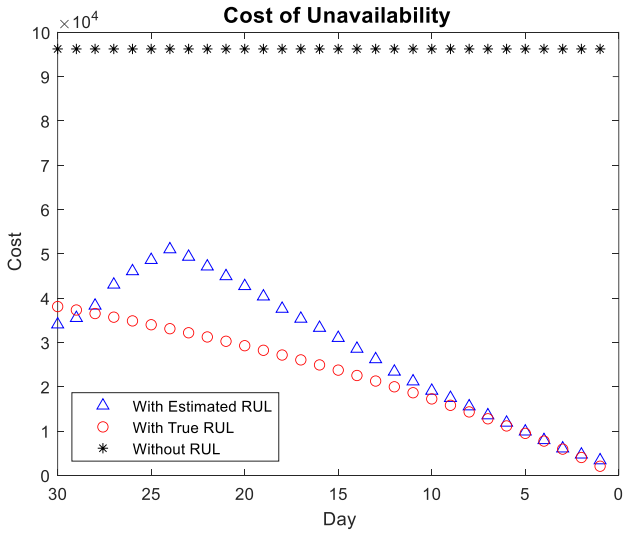


Figure 3-15. Wind Turbine cost of unavailability vs. remaining useful life considering only bearing faults, perfect maintenance and no limitation in shift times.

The figure shows that using combined RUL-based (condition-based maintenance (CBM)) and traditional reliability-centred maintenance (RCM) vs. RCM alone has between 60 to 100K GBP saving per year. It also shows that as one approaches the final days of the component, the gap in cost closes. It is also showing the direct relation between availability and the cost of unavailability.

Considering the imperfect maintenance probability with β from 0.1 to 0.4, the availability of the wind turbine can be demonstrated as Figure 3-16. By increasing the imperfect maintenance probability, the availability of the system will be decreased having both estimated and true RUL considered.

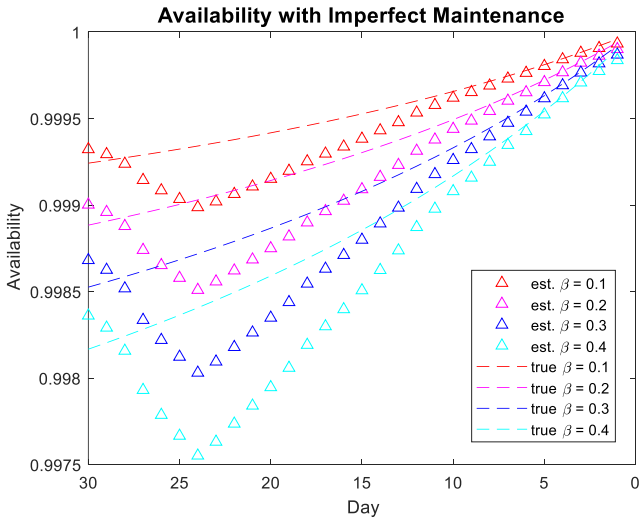


Figure 3-16. Wind Turbine Availability vs. remaining useful life considering only bearing faults, imperfect maintenance with different Beta factors and no limitation in shift times.

3.5.2 Example 2: Blade System failure

This example shows how to consider failures of more than one component in the proposed method. Figure 3-17 illustrates the FT of the blade system in the OWT. This FT has three repair and delay gates connected to intermediate events. In industrial applications, there are cases in which detailed information is spread among IEs and BEs. For instance, in this figure, failure rates are available for BEs level, but repair information is only available for IEs. In this case, BMMs should be created for IEs. It should be mentioned that, in this FT, the BE numbers have been kept the same as in (Kang, Sun and Guedes Soares, 2019) for readers to be able to follow FT construction, failure description and values with more details.

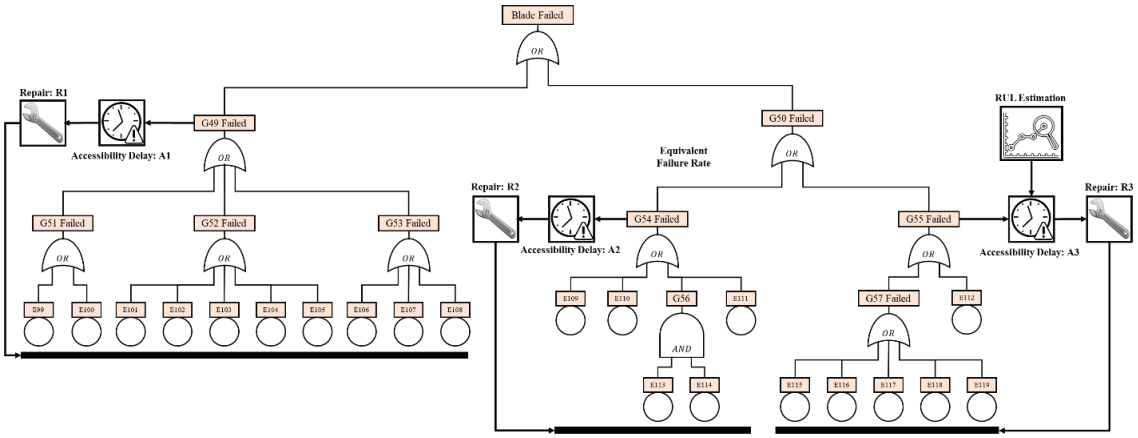


Figure 3-17. Sub-FT of blade system failure in the OWT

Regarding the above illustrated FT, the short description of twenty-one BEs' names and codes are shown in Table 3-3. This table also provides event names and codes for intermediate events.

Table 3-3. List of basic events for rotor system failure illustrated by FT in Figure 3-17 (Kang, Sun and Guedes Soares, 2019)

Event Code	Event Name	Event Code	Event Name
Basic Events			
E099	Open tip	E109	Clearance loosening at root
E100	Lightning strike on tip	E110	Cracks in the hub
E101	Cracks in the edge of blades	E111	Surface roughness in the hub
E102	Erosion in edges of blades	E112	Cracks in bearings of rotor
E103	Delamination in leading edges	E113	Mass imbalance in the hub
E104	Delamination in trailing edges	E114	Fault in pitch adjustment
E105	Debonding in edges of blades	E115	Corrosion of pins in bearings
E106	Delamination in shell	E116	Abrasive wear in bearings

E107	Crack with structural damage	E117	Pitting in bearings of rotor
E108	Crack on the beam-shell joint	E118	Deformation
		E119	Lubrication fault in bearings

Intermediate Events

G49	Structural fault	G54	Hub failure
G50	Bearing and Hub Failure	G55	Bearing's fault
G51	Tip damage	G56	Imbalance of blade system
G52	Edges damage	G57	Wear in bearings of the rotor
G53	Shell damage		

Figure 3-17, shows three repair gates in IEs G49, G54 and G55. From this tree, the following tree can be generated via substitution of these events with their corresponding BMM models.

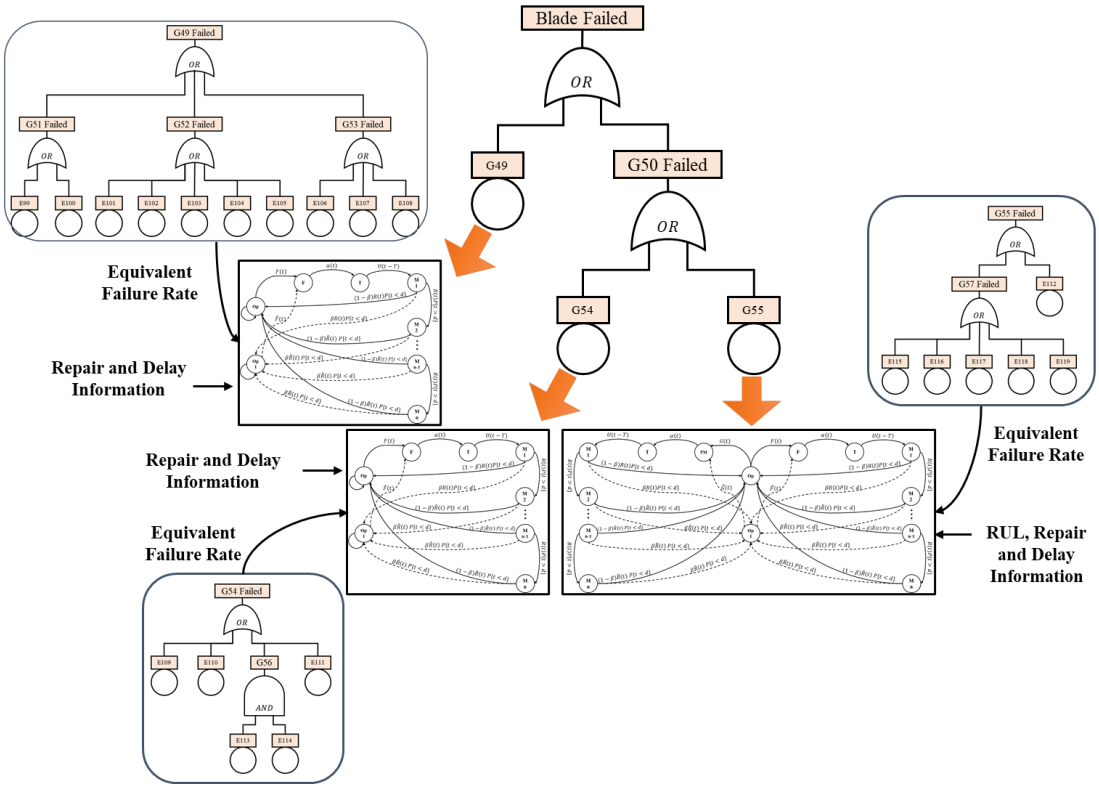


Figure 3-18. The procedure of forming complex basic events in Fault Tree of a blade system

To generate the FT in Figure 3-18 the following step should be taken.

A) when the repair gate pointing an intermediate gate, the whole sub-tree bellow that intermediate gate should be removed, and a its equivalent failure rate should be replaced. Thus, in this example three sub-trees bellow G49, G54 and G55 should be removed, and their equivalent failure rates should be replaced accordingly. Table 3-4

provides the equivalent failure rates for events G49, G54 and G55 calculated from data about BEs in the subtrees.

B) In the second step, repair rates, delay information, the probability of imperfect maintenance, etc. should be added to CBE box. In this example, three types of repair and weather delay information for four seasons were available. In addition, the RUL estimation is available from previous example that should be added to CBE of G55.

C) In the third step, the BMM models can be generated. As shown in this example, only one CBE has the BMM with RUL estimation. It is not realistic to expect RUL and many other parameters for all components in WT.

D) In the fourth step, the new FT with CBE will be formed. In this example, the provided FT of Figure 3-17 has been simplified to FT of Figure 3-18 with CBEs. In the procedure of creating the FT with CBE, there are two questions to be addressed: Why was sub-tree conversion required? Why are BMMs different?

Basic event G49 and G54 represent blades structural failure and hub failure respectively. It is assumed that their failures obey exponential distribution and there is no RUL estimation for these two events. So, the Markov model of Figure 3-4 is used for them. These two events have three types of repair rate including minor repair, major repair and major replacement provided in Table 3-4. For basic event G55, the Markov model of Figure 3-9 can be used in which both RUL-based failure and traditional one has been considered as explained in Example 1.

Table 3-4. List of basic events for blade system failure illustrated by FT in Figure 3-18. The procedure of forming complex basic events in Fault Tree of a blade system. (Faulstich, Hahn and Tavner, 2011b)(Carroll, McDonald and McMillan, 2016b)

Event Code	Event Name	Failure Rate	Repair Code	Repair Rate (Hours)	Model Type
G49	Blades structural failure	1.26E-5 Exponential	R1	Minor Repair: 9 Major Repair: 21 Major Replacement: 288	Figure 3-10
G54	Hub failure	2.74E-5 Exponential	R2	Minor Repair: 10 Major Repair: 40 Major Replacement: 298	Figure 3-10
G55	Bearings failure	5.25E-6 Exponential + RUL Estimation	R3	Major Replacement: 1000	Figure 3-9

Table 3-5. Waiting delay for appropriate weather window: Probability of facing inappropriate weather delay and its mean waiting time for each season (Santos, Teixeira and Soares, 2015)

Season	Probability of occurrence	Mean Waiting Time (day)
Winter	0.30	60
Autumn	0.50	30
Spring	0.60	10
Summer	0.80	3

Drawing from (Santos, Teixeira and Soares, 2015), the waiting delay for appropriate weather window can be expressed with two parameters including the probability of occurrence and mean waiting time for an appropriate weather window. Table 3-5 provides these parameters for different seasons. Values are used to compare the effect of different seasons.

Considering the weather delay parameters based on the above tables and using minor repair values, the result shown in Figure 3-19 is obtained. It is assumed that the maintenance is perfect. Figure 3-19 compares the average unavailability of the blade system for different seasons. As shown in example 1, the unavailability of the system without RUL estimation is higher than the other cases. Once more the accuracy of RUL estimation affects the average unavailability of the system. The accessibility of offshore wind turbines is limited during winter and this why winter has more average unavailability while summer has less average unavailability.

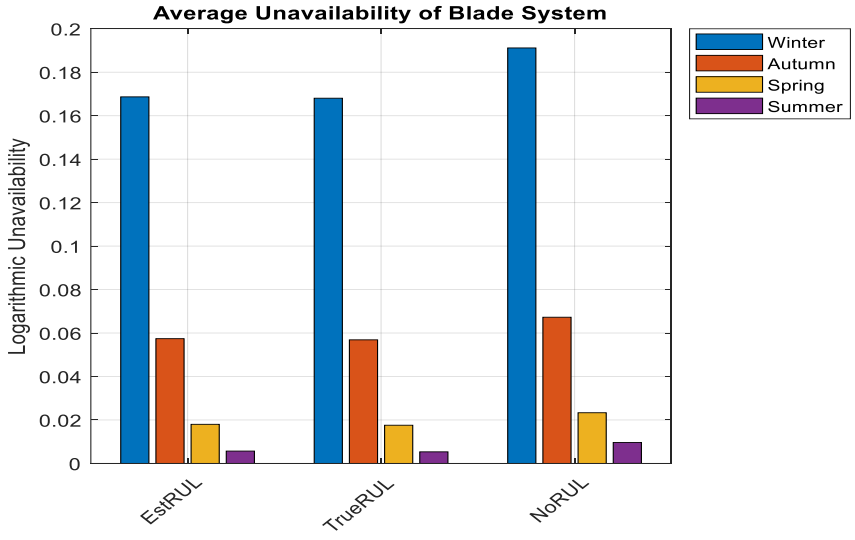


Figure 3-19. Average unavailability of blade system for different seasons considering minor repairs vs. (a) using estimated RUL, (b) using true RUL, and (c) without RUL.

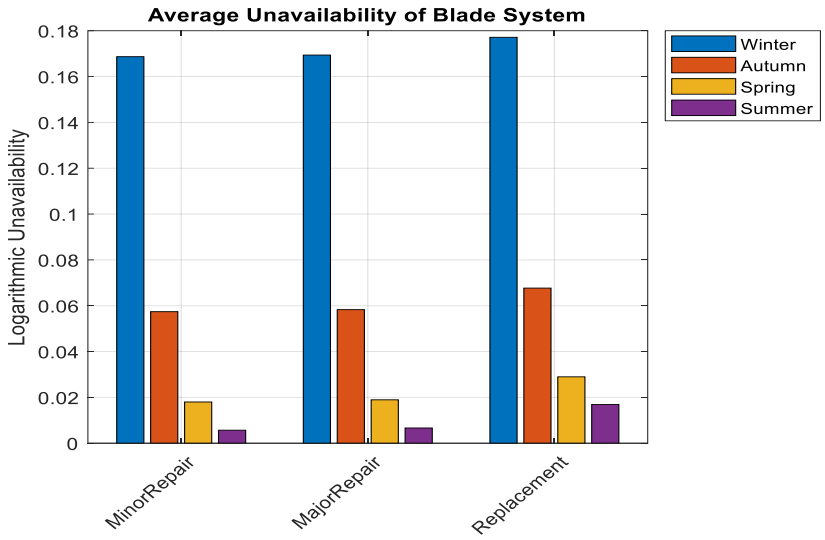


Figure 3-20. Average unavailability of blade system for different seasons considering minor repairs vs. (a) Minor Repair, (b) Major Repair, and (c) Replacement

Figure 3-20 shows, for different seasons, the effect of different repair/replacement values when the estimated RUL is used. It can be seen that replacement increases the average unavailability and reduces the difference between different seasons.

3.5.3 Example 3. Rotor System Failure

This example demonstrates the abilities of the proposed method in the steady-state availability analysis of OWT using a dynamic fault tree in conjunction with the BMM to model the system. It has been noticed that only a few research works have addressed the dynamic fault tree modelling of an offshore wind turbine. A sub-tree from (Zhang *et al.*, 2016) is shown in Figure 21 to illustrate dynamic fault tree modelling and its steady-state availability evaluation. Event codes have been kept as they were in the original thesis to enable readers find more details on DFT construction, failure description, etc. The figure shows the dynamic fault tree for rotor system failure. In this fault tree, there are four static gates, one dynamic gate (“cold spare gate”) and eight basic events.

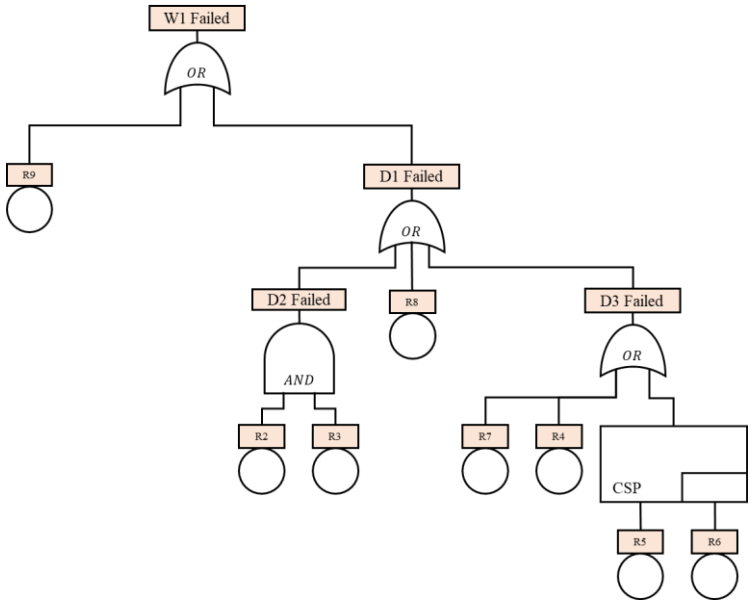


Figure 3-21. Dynamic sub-FT of Rotor system (Zhang et al., 2016).

Table 3-6 summarised event codes, event names and their failure rates. Regarding the event code R9, the output of example 2 will be considered as an input for this event. Actual repair rates are not available for Event R2 to R8, and for illustrative purposes, hypothetically the repair rates are considered as failure rates multiplied by 1000. Accessibility delays have been considered as equal for all components and the same as in example 2. Although assumed data is used for illustration, application would be identical on real data.

Table 3-6. List of basic events for rotor system failure with their failure rates (Zhang *et al.*, 2016)

Event Code	Event Name	Failure Rate	Event Code	Event Name	Failure Rate
Basic Events					
R2	First Encoder Failure	11E-6	R6	Power Unit 2 Failure	57E-6
R3	Second Encoder Failure	11E-6	R7	Limit Switch Failure	10E-6
R4	Lighting Protection Failure	10E-6	R8	Safety Cut-off	10E-6
R5	Power Unit 1 Failure	57E-6	R9	Blade Failure	Example 2
Intermediate Events					
D2	Encoders Failure		D3	Driver System Failure	
D1	Pitch System Failure		W1	Rotor System Failure	

Figure 3-22 compares the average unavailability of the rotor system when (a) estimated RUL (b) true RUL and (c) No RUL has been considered inside the event R9.

This figure also provides the average unavailability for different seasons. In general, the unavailability of the rotor system has the same pattern as that of the blade system. Also, the average unavailability of the rotor system is less than the average unavailability of the blade system.

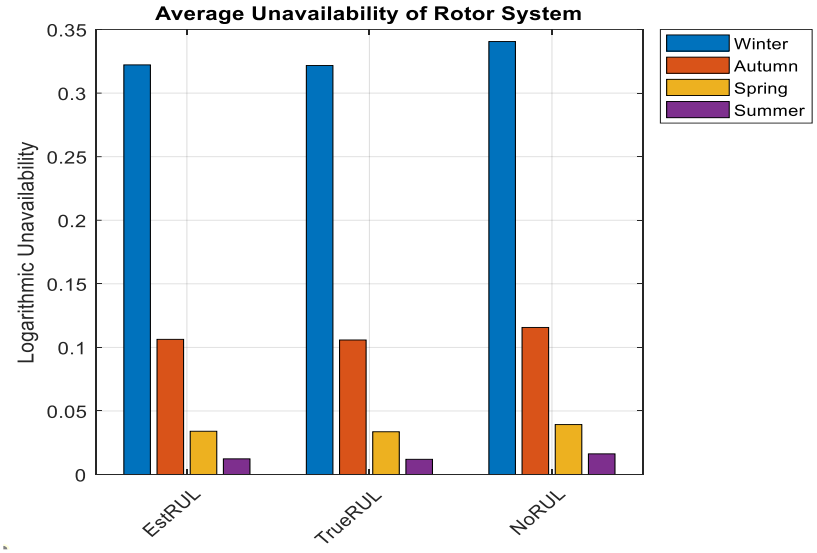


Figure 3-22. Average unavailability of rotor system for different seasons considering minor repairs vs. (a) using estimated RUL, (b) using true RUL, and (c) without RUL.

Figure 3-23 shows a similar pattern for the rotor system to that for the blade system, regarding the effect of different seasons on the unavailability under different repair and replacement policies. Once more estimated RUL is used for event R9, and as it can be seen, replacement increases slightly the average unavailability and reduces the difference between different seasons. It is clear that the rotor system has less average unavailability in comparison to the blade system.

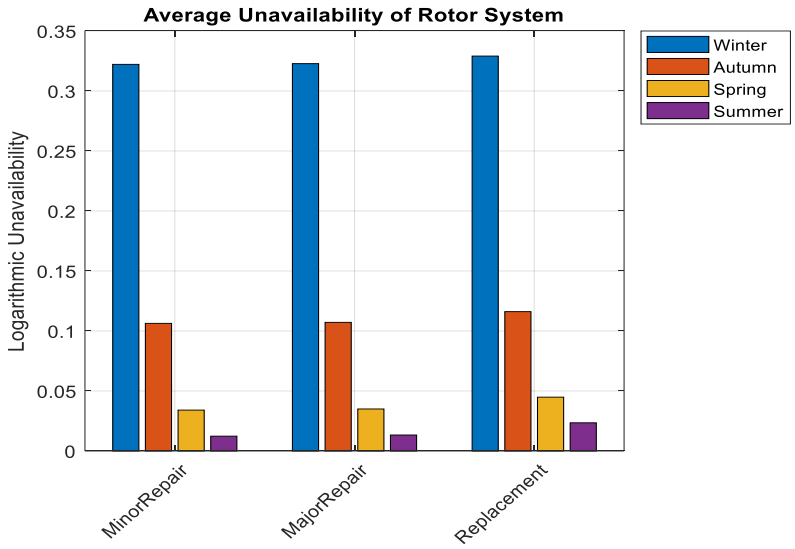


Figure 3-23. Average unavailability of blade system for different seasons considering minor repairs vs. (a) Minor Repair, (b) Major Repair, and (c) Replacement.

Importance measures can be used for criticality analysis in a Fault Tree. Table 3-7 provides the importance measure values of basic events in this example for different seasons when estimated RUL and minor repair are considered. As can be seen in Winter, Autumn and Spring, R9 was the most critical component contributing most to the OWT's unavailability, while in Summer R5 and R6 have more criticality. This table can be produced for different other situations (e.g. considering true RUL and Major repair).

Table 3-7. Importance Measure vs different seasons (considering estimated RUL and minor repair)

Event Code	Importance Measure (Considering Estimated RUL and Minor Repair)			
	Winter	Autumn	Spring	Summer
R2	0.006164045102912	0.018675833863331	0.058308442794917	0.160873862708873
R3	0.006164045102912	0.018675833863331	0.058308442794917	0.160873862708873
R4	0.005885797863490	0.017832799925451	0.055676378465111	0.153611957994260
R5	0.018909606632872	0.057292356885840	0.178874375222576	0.493517067209361
R6	0.018909606632872	0.057292356885840	0.178874375222576	0.493517067209361
R7	0.005885797863490	0.017832799925451	0.055676378465111	0.153611957994260
R8	0.005885797863490	0.017832799925451	0.055676378465111	0.153611957994260
R9	0.523479877021890	0.539743356626457	0.528234959485040	0.460238368615908

3.6 Capabilities and Limitations of the Proposed Method

In this section, some of the capabilities and limitations of the proposed method are discussed. A number of suggestion are given to overcome limitations.

3.6.1 Capabilities of the Proposed Method

- A novel butterfly maintenance model is proposed that considers both traditional and data-driven availability analysis by combining both simple failure distribution and remaining useful life (RUL)-based transitions in a single MRGP. The proposed model can utilize both condition-based maintenance and reliability-centered maintenance.
- The introduced BMM model can incorporate more advanced logistics delays, weather delays and daily briefing and transfer delays. Consideration of these accessibility and delay factors will lead to a more realistic availability evaluation of the offshore wind farm.
- In the Markov modelling theories, the issue of considering the time-based deadline for a transition has rarely been discussed. This chapter used the deadline transitions to model shift time limits of the O&M for offshore wind turbines.
- Imperfect maintenance for BMM has been introduced that can make the model and its evaluation more realistic. The proposed model is not only able to consider

imperfect maintenance but also able to use this option to model the life extension of the turbine.

- As shown in the examples, the presented model is capable of modelling non-exponential system behaviour using MRGP. The MRGP has higher computation complexity but at the same time offers more power in modelling complex failure behaviour.
- The use DFT has been proposed as a means of combining the availability models of components and evaluating the total availability of the system. Using DFTs provides an intuitive perspective on fault propagation in the system. It should be mentioned that in the literature, models with this level of details are usually considered a component of the system while in this chapter, the combination of such models is also discussed.

3.6.2 Limitations of the Proposed Method

- The proposed model is not capable of handling common cause failures. Consider we have two components A and B and assume a common cause failure can occur for them. In this case they cannot have a separate Markov model and instead they need to have a joint model to be able to consider the effect of the common cause failure in the model.
- The BMM model proposed can provide an exact value for availability. However, combining the BMM model with the hierarchical dynamic fault tree solution (from Chapter 2) cannot provide the exact value for the availability and the calculations will only be approximate. This issue can be solved by improving the solution provided for hierarchical dynamic fault tree and its availability evaluation.
- Due to the lack of data, the thesis did not examine in practice the use of deadline transitions to model shift time limits of the O&M for offshore wind turbines; this remains as the future work.
- The BMM model and the availability evaluation are highly dependent on the correctness of RUL estimation. Thus, having a poor estimation can lead the model to provide wrong values for availability.

3.7 Conclusion

An advanced butterfly maintenance model for offshore wind turbines has been provided which considers non-exponential distribution functions and the effect of a range of conditions. The model enables merged condition-based maintenance and reliability-

based maintenance, and this combination enables fine-tuning of maintenance and increase in system availability. In case of abrupt fault types or unavailable RUL estimation, it is possible to put zero for G(t) transition and practically ignore the left part of the model. So, the model can be still useful in those cases.

To create a data-driven model, a real-time transition based on estimated RUL has been addressed. Considering real-time transitions in MRGP is a novel feature in the field.

The model has a comprehensive perspective regarding the different types of logistics delays, weather delay, and shift time limitations and provides imperfect coverage which is rare in the availability and maintainability modelling of offshore wind farms. In addition, it captures imperfect maintenance and life extension. These features together with the use of RGMP to model non-exponential system behaviour and the ability to merge with dynamic fault trees makes the approach novel and powerful.

To enable use of the BMM model as complex basic event in FTs or DFTs, no interaction must be assumed between any two BMMs. Correlated or interactive failures must be modelled within separate models, and this can make results approximate and less accurate in such cases.

We have shown that the proposed Butterfly Maintenance Model can be used for the analysis of condition-based maintenance, reliability centred maintenance or both. The potential benefits were illustrated through numerical results on examples drawn from OWT technology. The uncertainty and sensitivity analysis of the proposed model can be further established in the future using perturbation theory (Do Van, Barros and Bérenguer, 2008). Based on the idea mentioned in (Gheraibia *et al.*, 2019), it is possible to use artificial intelligence to update the availability model and make it more adaptive.

Chapter 4

4 Addressing Unpredictability in Dependability of Open Multi-Agent Systems

Open systems of systems are unpredictable and can form large numbers of configurations which would be impossible to enumerate exhaustively and analyse for reliability before deployment. In this chapter, we propose a way of dealing with this uncertainty by making the reliability model of a system executable during operation. In operation, much of the uncertainty is indeed resolved, for example a configuration of a multi-agent system, and conditions of health of its systems are known. The proposed model is a fault tree augmented with complex basic event described as Markov model (see also chapter 2 for complex basic events). The model provides can be linked to the monitoring and diagnosis capabilities of a system and provides a real-time reliability evaluation of the system. In order to explore this proposal, a model has been developed

and implemented on a real industrial case study on drones for power network inspection in collaboration with the KIOS Research and Innovation Centre of Excellence in the context of the EU SESAME project. One outcome of this chapter is the **SafeDrones** Python package that provides an implementation of this work with many reusable elements.

4.1 Reliability & Unmanned Aerial Vehicles Industry

There are many potential applications for Unmanned Aerial Vehicles (UAVs), including logistics, emergency response, filming, traffic monitoring, search and rescue, rail surveillance, and infrastructure inspection (Christine M. Belcastro *et al.*, 2017). However, one of the major barriers to widespread deployment and acceptance of UAVs is that of safety, particularly for operations in urban areas where UAV failure brings a higher risk of harm. For instance, during testing for Amazon's planned fast drone-based delivery service, their drones crashed five times over a four-month period in 2021 (Amazon Drone Crashes, Delays Put Bezos's Delivery Dream at Risk - Bloomberg). Therefore, safety and reliability must be key objectives during both the design and operation of UAVs to help minimise risk and improve likelihood of mission success (Sadeghzadeh, Mehta and Zhang, 2011). When it comes to UAVs, there are two major viewpoints for reliability evaluation: time and distance.

Time-based reliability is the probability that the UAV carries out its mission during a specific period of time. Distance-based reliability is the probability of correct functioning of vehicle at a specific distance from its base station. In addition, reliability can also refer to the likelihood of the UAVs crashing into the ground or experiencing other mission failures. Reliability can be defined generally as the probability of a system functioning correctly over a given period of time (Trivedi and Bobbio, 2017). In this chapter, the main focus is to evaluate the time-based reliability of UAVs having one or set of symptoms from UAV's monitoring system.

There is a long history of reliability engineering techniques intended to help analyse, understand, and prevent failures. Among the most popular are Fault Tree Analysis (FTA) (Stamatelatos *et al.*, 2002) and Failure Modes and Effects Analysis (FMEA). Such techniques were originally manually applied but over time have evolved and now form integral parts of comprehensive, tool-supported methodologies, encompassed under Model-Based Safety Analysis (Sharvia *et al.*, 2016). Using such

approaches during the design of a UAV, it is possible to determine the ways in which it can fail and the likelihood of those failures.

UAVs must often operate independently in dynamic, unpredictable environments with varying mission goals, all of which are difficult to capture in a design-time analysis model. By combining design-time knowledge with safety monitoring applied at runtime, we can perform dynamic reliability evaluation and obtain a clearer picture of UAV reliability during operation.

In the area of UAVs and their real-time health or remaining useful life estimation, there are some existing research works. For instance, an Integrated Vehicle Health Management (IVHM) framework for decision-making system has been introduced by (de Medeiros *et al.*, 2014). In this thesis, the remaining useful life (RUL) of the UAV is estimated through fault tree analysis and has been used for health-based task allocation in UAVs. In (Shi, Yang and Quan, 2016), the reliability of UAVs with different number of rotors and different configurations has been evaluated based on propulsion system's health and its real-time controllability status.

Reference (Barr *et al.*, 2017) considered the subject of ground collision in low-attitude airspace and introduced a third-party casualty estimation. It also presented a preliminary risk analysis approach based on standard risk analysis and probabilistic model-based approach for UAVs. Meanwhile, (Christine M Belcastro *et al.*, 2017) provided experimental flight test techniques with different possible set of hazard-based test scenarios to evaluate the safety of sUAS operations. The hazard analysis of sUAS mishaps and the future hazard analysis of a collection of sUAS use cases has been summarized and identified by (Christine M. Belcastro *et al.*, 2017).

Although there has been research into the reliability assessment of UAVs using FTA, none have adopted a compositional approach to the system component hierarchy that also considers the dynamic behaviour of the propulsion subsystem. This thesis, for the first time, constructs the fault tree model of UAV from its recognized operational faults and evaluates its reliability from this model. This thesis considered the dynamic behaviour of the propulsion system through Markov theorem and uses the results as a basic event in the fault tree. The constructed fault tree is useful at design time to enable designers to identify and address weaknesses in the design, but importantly it is also usable at runtime to diagnose faults and evaluate their effects on the system resiliency during operation.

This thesis proposes a new model-based approach to improve reliability and safety of UAVs called SafeDrones. SafeDrones builds upon static design-time knowledge in the form of fault trees by combining them with dynamic Markov-based models and real-time monitoring to perform continuous reliability evaluation at runtime. The result is a modular safety monitor known as an Executable Dependable Digital Identity (EDDI), which can then be used to inform operational decision making.

4.2 Reliability Modeling using Arrhenius Equation

The lifetime reliability of a processing unit has a strong correlation with its temperature (Ottavi *et al.*, 2014). Moreover, a processor's temperature depends on the UAV's performance and utilization. To capture this interaction between reliability and temperature, the Arrhenius equation has been used. The Arrhenius equation is used to compute the *MTTF* acceleration factor *(AF)* depending on the processor's actual and reference temperatures.

$$AF = e^{\frac{E_a}{k}\left(\frac{1}{T_r} - \frac{1}{T_a}\right)} \tag{4-1}$$

where E_a is the activation energy in electron-volts, k is Boltzmann's constant (8.617E-05), T_r is the reference temperature and T_a is the actual temperature. The acceleration factor *(AF)* is then used by the MTTF model to evaluate the effects of temperature on the MTTF. The final MTTF of the processor is calculated using the following equation:

$$MTTF = \frac{MTTF_{ref}}{AF} \tag{4-2}$$

where $MTTF_{ref}$ is the reference MTTF, estimated at the reference temperature. $MTTF_{ref}$ is usually given by the system's designers.

4.3 The Executable Digital Dependable Identity (EDDI)

Despite existing standards and guidelines, there is a great deal of variation in how assurance of dependability attributes is realized and claimed for concrete systems. This makes it difficult for third parties like certification authorities to analyse and evaluate the assurance approach in general, and especially when the systems are to be open, adaptive, or autonomous, like platooning cars (Kabir, S., et al., 2019).

To overcome this issue, Digital Dependability Identities (DDI) were created (Schneider *et al.*, 2015a; Wei *et al.*, 2017; Armengaud, D Schneider, *et al.*, 2021). A DDI is a structured, modular, and hierarchical model of a system's dependability properties. An assurance case is at the heart of the DDI, arguing for the assurance of the appropriate

dependability attributes and connecting all models and artefacts (e.g. requirements, assumptions, architectural models, dependability analyses, evidences) essential for the argumentation. A DDI is created and updated throughout the design process, issued when the component or system is launched, and then maintained during the component or system's lifespan. DDIs are utilized for the hierarchical integration of systems to "systems of systems" in the field, as well as the integration of components to systems during development.

An Executable Digital Dependability Identity (EDDI) is an extension of the DDI concept that is intended to be executable at runtime. It leverages the design-time dependability models stored in the DDI and augments them with event monitoring and diagnostic capabilities to provide real-time feedback on reliability, security, and safety issues, thereby supporting safe operation and dynamic dependability management. Importantly, EDDIs are intended to act cooperatively when applied within a distributed multi-robot or multi-agent system, enabling on-the-fly reconfiguration, communication, and adaptation. The idea is to support dynamic adaptive system assurance and dependability management through event monitoring, run-time diagnostics, risk prediction, and recovery planning.

Like DDIs, EDDIs are based on the Open Dependability Exchange metamodel (Sorokos, 2021). An EDDI generally consists of some higher-level ODE-based system models for diagnostics, capability (e.g. success trees) and risk prediction (e.g. fault trees, Bayesian networks) and lower-level models for event monitoring and reliability estimation (e.g. Markov models, Bayesian networks). Once connected to sensor data and other pertinent system information, the EDDI can use these models to perform calculations to provide feedback and recommendations to the host system. SafeDrones can be considered as a prototype instantiation of the EDDI concept that utilises fault trees and SMPs to provide dynamic reliability evaluation for UAVs.

Executable Scenarios (ExSce) are model-based narrative descriptions of missions guiding the engineering of open adaptive applications. An ExSce supports the definition of both mission-relevant and mission-plausible information, including but not limited to, the environment and its dynamics, time and events, objects (e.g., inspected building) and subjects (e.g., human operators) and their potential behaviour. Executable Digital Dependability Identities (EDDI) is an extension of DDI, that are model-based artifacts spanning the adaptive system life cycle. An EDDI carries verifiable dependability models of its reference system produced at design time based on ExSec, capturing safety and

security hazards, their causes, effects, and possible corrective measures. The joint EDDI and adaptive system execution enables dynamic adaptive system assurance and dependability management through event monitoring, run-time diagnostics, risk prediction, and recovery planning.

4.4 SafeDrones Approach

SafeDrones is an approach for real-time reliability and risk evaluation of multi-robot (multi-UAV) systems. The main goal of this work is to develop an early prototype instantiation of the EDDI concept for runtime reliability estimation for UAVs. It makes use of fault trees as the overall model with CBEs to support dynamic evaluation. A fault tree consisting of 9 main failure categories and 28 basic events is proposed for a generic UAV in Figure 4-1. However, to simplify the explanation of the methodology, a smaller fault tree of the UAV is provided in Figure 4-2.

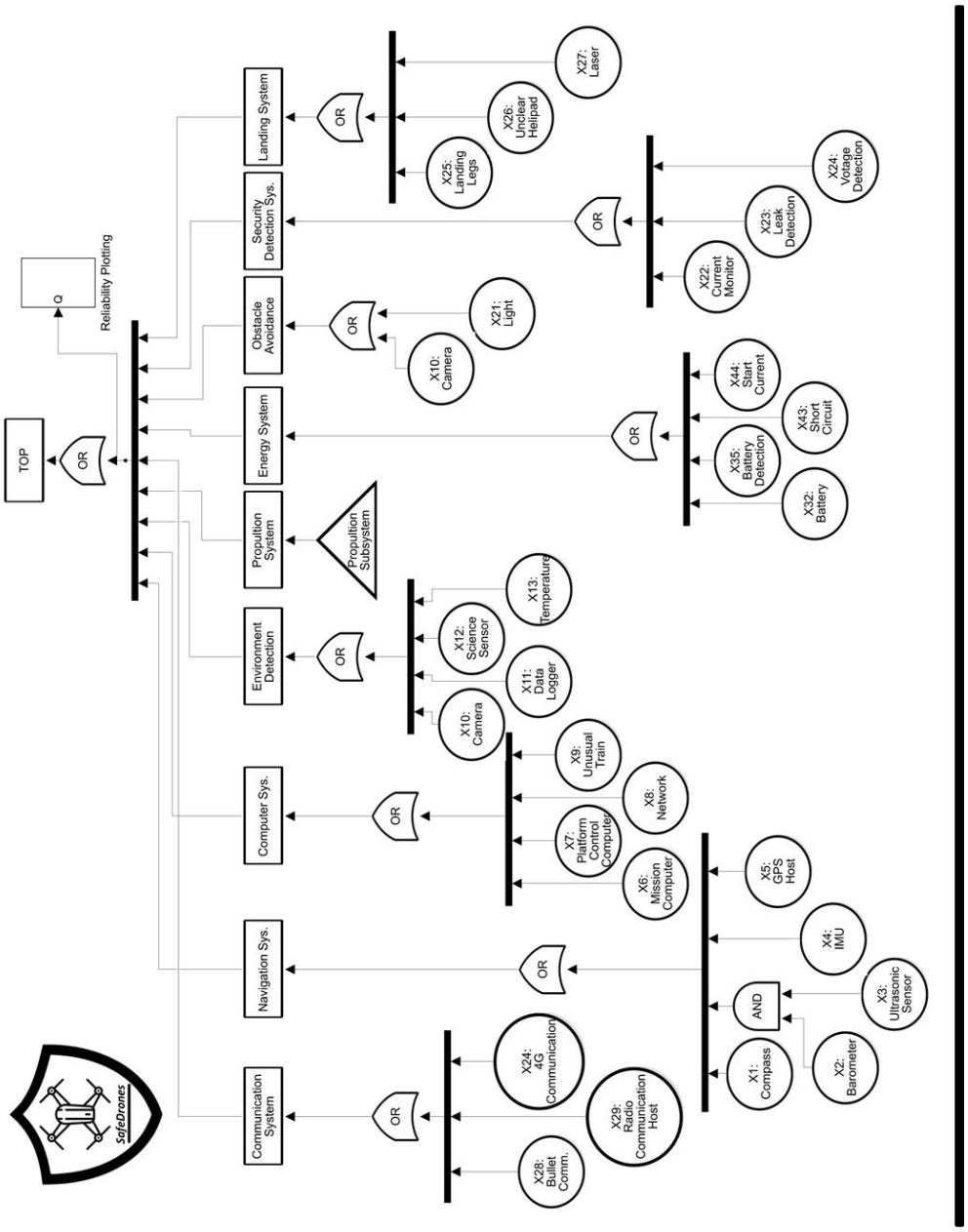


Figure 4-1. A fault tree of UAV consists of nine main failure categories and twenty-eight basic events.

The contribution and capabilities of the SafeDrones approach are as follows: 1) SafeDrones expands the idea of FTA with CBEs to not only consider SMPs but also other evaluation functions like the Arrhenius Equation; 2) it proposes the idea of having symptom events for each CBE; 3) it is also able to handle reliability evaluation of reconfigurable systems by using pre-defined models in one CBE (e.g. consider a hexacopter capable of reconfiguring its propulsion system on-the-fly from PNPNPN configuration to PPNNPN configuration), and 4) finally, SafeDrones provides Python functions which can be executed on each UAV and provide real-time reliability and MTTF evaluation. This thesis primarily explores the first and fourth capabilities.

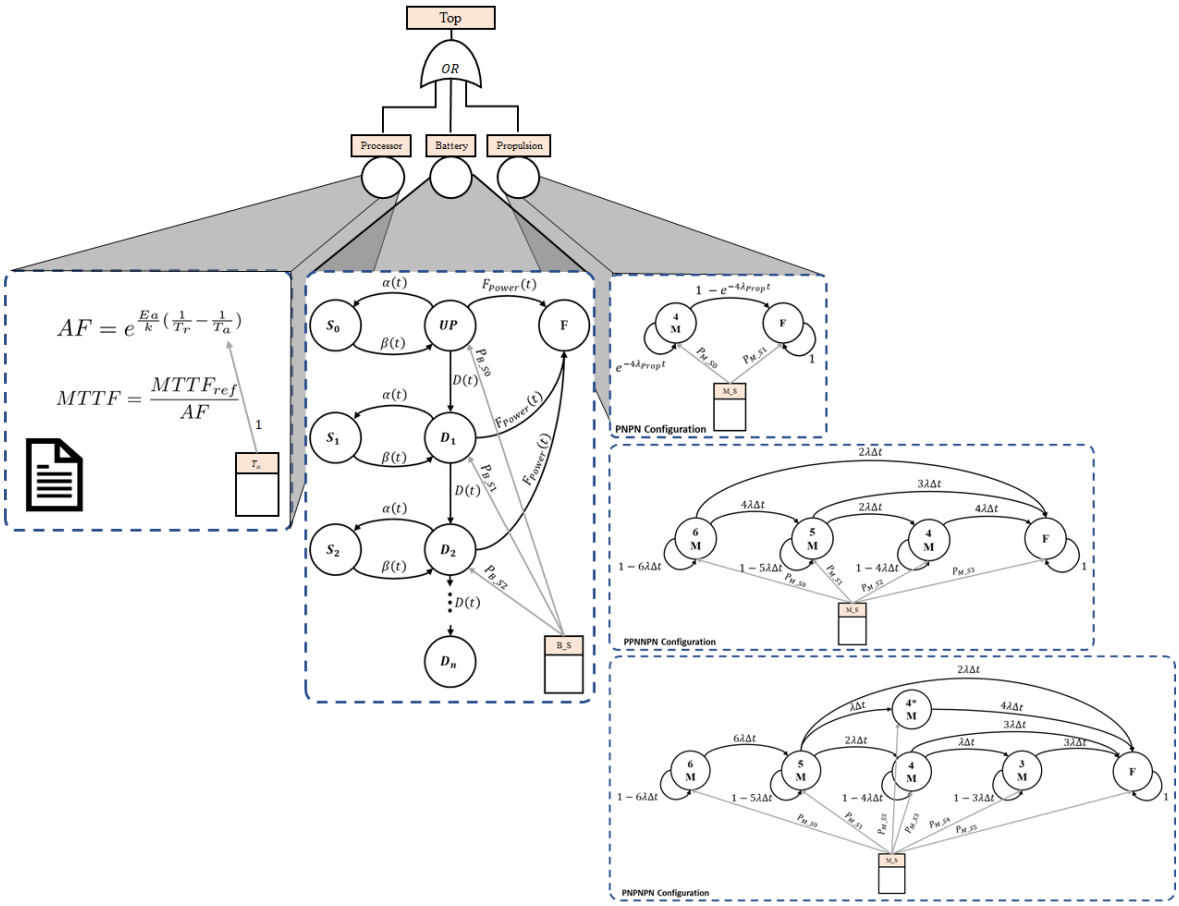


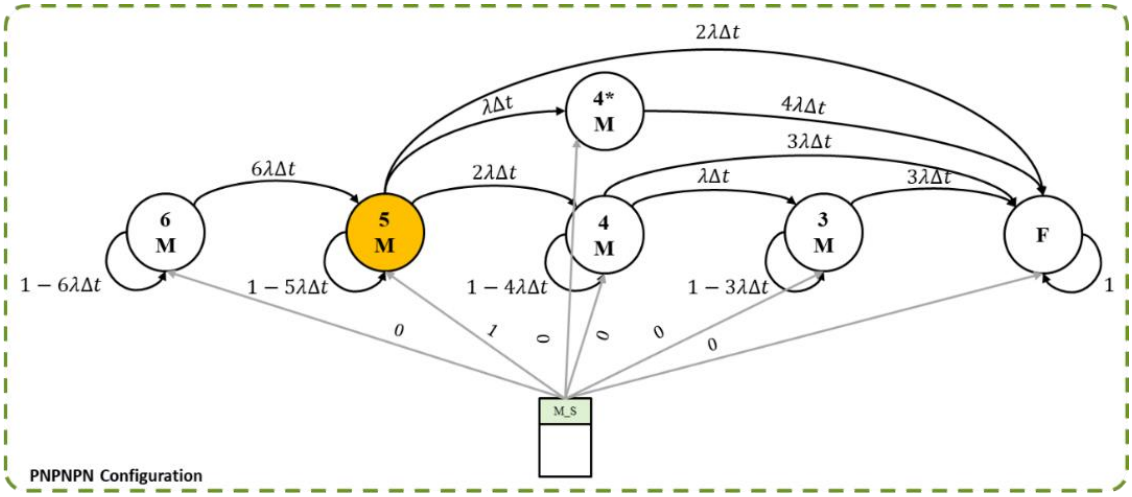
Figure 4-2. Small FTA of a UAV considering complex basic events with failure symptoms and three different types of propulsion system reconfiguration.

The tree provided in Figure 4-2 has three CBEs for battery failure, propulsion system failure and processor failure.

The processor failure has a symptom of actual temperature (T_a is the symptom) and based on the Arrhenius Equation (in previous section), the reliability and the MTTF values of this basic event can be updated during the mission. The idea can be implemented for any component in the robot where its reliability can change based on temperature variation. The middle CBE is for the battery failure. This model is provided by (Kim, Ghosh and Trivedi, 2010) and considers battery degradation as well as failure. In this thesis, we have used the battery model with four degradation levels and the battery level status *B_S* is included as a symptom. So, based on the battery level status, the initial probability vector in the SMP will be updated and then the probability of failure (unreliability of the battery) will be updated accordingly.

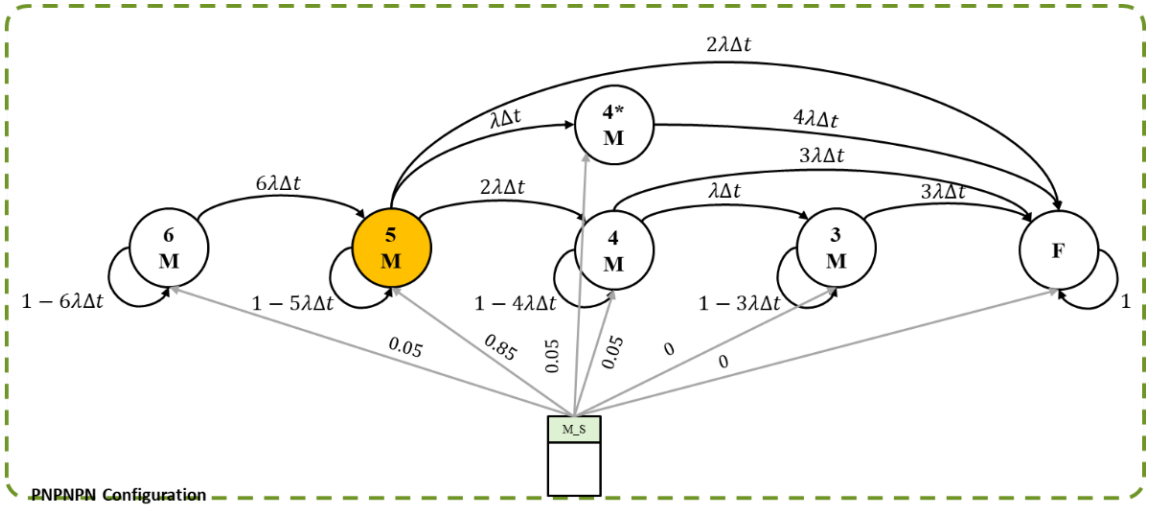
The third CBE is a propulsion system failure. The CBE is chosen to show the capabilities of SafeDrones for handling system reconfiguration. The first configuration is for a quadcopter that has two propellers rotating clockwise (P) and two propellers rotating anti-clockwise (N) forming PNPN configuration. The second and the third configurations

considered for hexa-copters with two different PNPNPN and PPNNPN configurations (P stands for clockwise rotation and N stands for anti-clockwise rotation). The detailed construction and simplification of these models has been discussed in one of my previous research projects (Aslansefat, Marques, *et al.*, 2019).




```
P_Fail, MTTF = Motor_Failure_Risk_Calc([0,1,1,1,1,1], 'PNPNPN', 0.001, 100)
print(P_Fail)
print(MTTF)
```

Figure 4-3. Markov model of a hexacopter with PNPNPN configuration and motor status (*M_S*) as a symptom – binary link between the symptom and the system's states.




```
P_Fail, MTTF = Motor_Failure_Risk_Calc([0,1,1,1,1,1], 'PNPNPN', 0.001, 100)
print(P_Fail)
print(MTTF)
```

Figure 4-4. Markov model of a hexacopter with PNPNPN configuration and motor status (M_S) as a symptom – considering the uncertainty of symptoms.

Figure 4-5 and Figure 4-6 further illustrate the idea of merging real-time monitoring and diagnosis with FTA. In a traditional FTA, the tree consists of a top layer, a number of intermediate layers, and a basic events layer. However, in SafeDrones there is a new layer called the symptoms layer. In the symptoms layer, the safety expert(s) should identify the potentially observable events in the system and define the relation between symptoms and basic events. For instance, in Figure 4-2, the symptoms are temperature, battery status, and motor status along with motor configuration. In Figure 4-2, it is assumed the temperature symptom only affects the processor and has no effect on the others. In this proposed reliability modelling approach, it is recommended to use CBEs to link with the symptoms. A CBE can take many forms, e.g., a multi-state Markov chain where the symptom affects its current state, a Bayesian Network where a symptom can form a belief, or some other reliability function where a symptom can be a parameter on it, etc. The link between symptoms and basic events can be both deterministic and probabilities values.

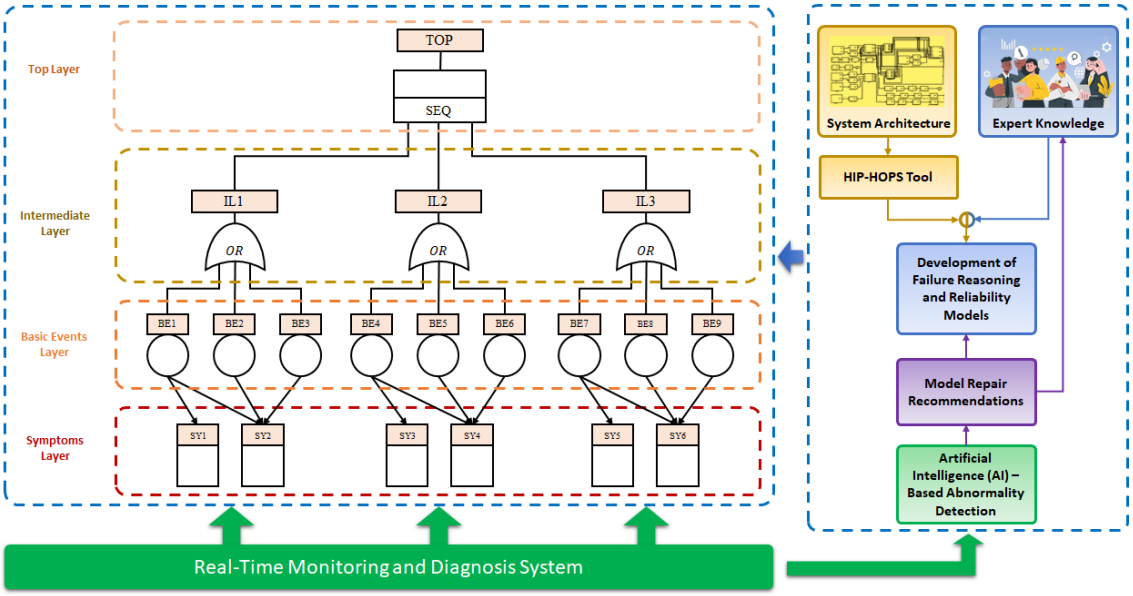


Figure 4-5. Overall view on merging real-time monitoring and diagnosis system with Fault Tree Analysis (right-side focusing on the way that DFT is generated).

In the right section of above figure, we propose an approach to combine systematic safety model generation and the expert knowledge. In addition, we show how the data from real-time monitoring and diagnosis system can be utilized for having a model repair recommendation. To read more about the idea, it is suggested to read our research on fault tree repair recommendation (Gheraibia *et al.*, 2019). On the other hand, in the following paper, the right-side of figure is focused on the idea of safety models being executable.

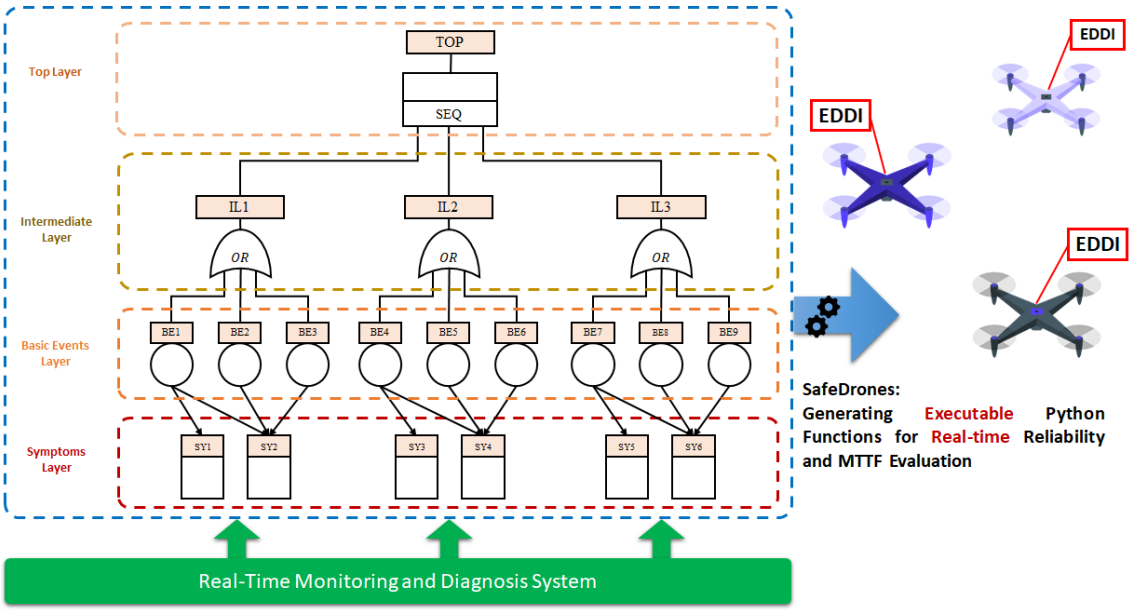


Figure 4-6. Overall view on merging real-time monitoring and diagnosis system with Fault Tree Analysis (right-side is focusing on the way that EDDI can be generated for UAVs).

As discussed in the previous EDDI Section, the EDDI concept uses real-time evaluation of dependability attributes like reliability as function(s) to update the mission accordingly as part of a dependability-driven decision-making system. This could lead to a variety of responses, such as reconfiguration during the mission (e.g., switching a hexacopter to quadcopter mode in the event of possible motor faults), changes to mission parameters (e.g., emergency landing or return to base sooner), or even requests for predictive maintenance of affected parts. In SafeDrones, all the calculations are implemented in Python (available in the GitHub repository as mentioned in the first section) for runtime execution. The results could also be used by technologies like ConSerts (Schneider and Trapp, 2013) to generate conditional guarantee outcomes and provide the final decision accordingly. Moreover, based on the idea provided by (Gheraibia *et al.*, 2019), it is possible to investigate the use of monitoring data to obtain safety model repair recommendations.

4.5 Experimental Implementation

To evaluate the reliability models presented in the methodology section of this chapter, ICARUS toolkit is used (Savva *et al.*, 2021), which uses vision-based UAV monitoring platforms to automate the inspection of medium voltage power distribution networks. As Figure 4-7 shows, the UAV gathers data and provides a real-time data processing to identify poles and record their accurate positions. An off-the-shelf four-rotor UAV (DJI Matrice 300 RTK) equipped with different sensors, including temperature sensors, is used. On the top of the UAV, an NVIDIA Jetson Xavier NX

embedded platform was mounted to run the deep learning and navigation algorithms, allowing the UAV to perform inspection procedures autonomously. Additionally, the UAV is equipped with the SafeDrones tool, which monitors parameters such as processor temperature, battery level and execution time to estimate UAV reliability. Furthermore, SafeDrones can recommend actions like mission abort and emergency landing if the estimated reliability falls below a predetermined threshold. For the pole detection task, the UAV flew at a constant height of approximately 50m above the ground with the camera turned downwards. To identify poles (top-view) in videos under different background and lighting conditions, the tiny-You-Only-Look-Once (tiny-YOLO) v4 was employed.

Figure 4-7. Inspection procedure using ICARUS toolkit (Savva *et al.*, 2021) for pole detection.

Table 4-1. Input values for the parameters used in the models.

Parameters	Description	Setting
	Motor Parameters	
MC	Motor Configuration	PNPN (P: Positive clockwise direction, N: negative anti-clockwise direction)
Motor λ	Motor Failure Rate	0.0010
	Battery Parameters	
Battery λ	Battery Failure Rate	0.0001
D	Battery Degradation Rate	0.0064
α	Battery Usage Rate	0.0080
β	Battery Inactivity Rate	0.0070
	Processor Parameters	
u	Utilization	1.0000
$\{MTTF\}_{\{ref\}}$	Reference MTTF	1000 hours
E_a	Boltzmann Constant	8.617E-05
k	Activation Energy	0.3 electron-volts
T_r	Reference Temperature	$\{29\}^{\circ C}$

For our analysis we monitor processor temperature and battery level every 1 second to estimate the probability of failure for the UAV using the models described earlier. All the other input parameters are shown in Table 4-1. When the estimated probability of failure exceeds a specific threshold (we use 0.9 as a threshold for this analysis), an emergency action is taken. In this case, the action is to perform a safe emergency landing and continue the mission with another UAV. Note that the threshold can vary, depending on the mission and the time needed to safely land the UAV. The total execution time for the fault-free inspection mission to detect all the poles is 750 seconds. To demonstrate the SafeDrones concept, we use two scenarios:

- Fault-free scenario. In this scenario, all the components work properly without experiencing any faulty conditions.
- Faulty scenario. In this scenario, the battery stops working properly at a specific time X causing a sharp drop in the battery level and at time Y, where Y>X, the processor starts overheating due to unexplained circumstances. For this analysis, X equals 250 seconds and Y equals 400 seconds.

4.6 Experimental Results

This section reports the reliability analysis results for the two scenarios described in the previous Section.

4.6.1 Reliability Analysis of the Fault-Free Scenario

We first evaluate the probability of failure of the different components (battery and processor) and the total UAV for the fault-free scenario. It is assumed that the mission is about 800h. Figure 4-8 (a) and (b) show the battery level and processor temperature respectively (collected from UAV's telemetry logs), while Figure 4-8 (c) and (d) show the failure probability and MTTF for each component as well as the overall UAV. As Figure 4-8 (c) shows, the lower the battery level, the higher probability of failure. The sharp increase here when the battery level goes below 75% is because our model discretizes the battery level into four states (25% each), resulting in a jump when each discrete state is reached.

Additionally, Figure 4-8 (c) shows that the processor's probability of failure is also related to the UAV's cumulative processing time. The exact correlation between reliability and processor temperature is shown in Figure 4-8 (i), which illustrates how the processor's MTTF changes according to the current temperature. As is clearly shown,

when the processor's temperature increases, the MTTF also decreases. Finally, as can be observed in Figure 4-8 (c), the overall UAV failure probability does not exceed the 0.90 threshold for emergency action, indicating that the inspection mission was completed successfully. Note that the threshold value should be determined by a team of safety experts.

4.6.2 Reliability Analysis of the Faulty Scenario

In the first scenario, the overall probability of failure was satisfactory throughout and the UAV managed to complete the mission safely. However, it is also possible for faults to develop in any component, and so in the second scenario we investigate how the probability of failure can be changed by simulating a faulty battery and the processor overheating. Here the battery stops working properly at the 250th second. At this point the battery level drops sharply from 80% to 40% as Figure 4-8 (e) shows. The processor's temperature also suddenly increases at the 400th second.

Figure 4-8 (g) and (h) depict the impact of these simulated faults on the probability of failure and MTTF respectively. As Figure 4-8 (g) shows the failure probability threshold of the UAV is exceeded at the 500th second. This leads to an emergency landing of the UAV even if the mission was not completed. In a multi-UAV scenario, another UAV can be dynamically tasked to continue and complete the mission in this case.

The results highlight the benefits of both the proposed SafeDrones approach and the overall EDDI concept in helping to avoid dangerous accidents caused by failures. By combining safety analysis models and reliability functions executable at runtime, we can obtain a more comprehensive overview of UAV dependability during real-time operation, one that considers multiple subsystems and sensors as well as predefined thresholds and corresponding mitigating actions. Such an approach is particularly valuable for autonomous platforms where there is no human operator to monitor safety directly.

Consider the following figure and a scenario in which three UAVs are tasked to scan three fields in parallel when an internal fault occurs in UAV #3. Although it does not cause immediate failure, SafeDrones re-evaluates the reliability at runtime and determines it increases the risk above a dangerous threshold, so in order to reduce the risk of collision or spraying out of bounds, UAV #3 activates a fail-safe mechanism and returns to base. If UAV #2 has required availability, its mission can then be updated to cover the third field.

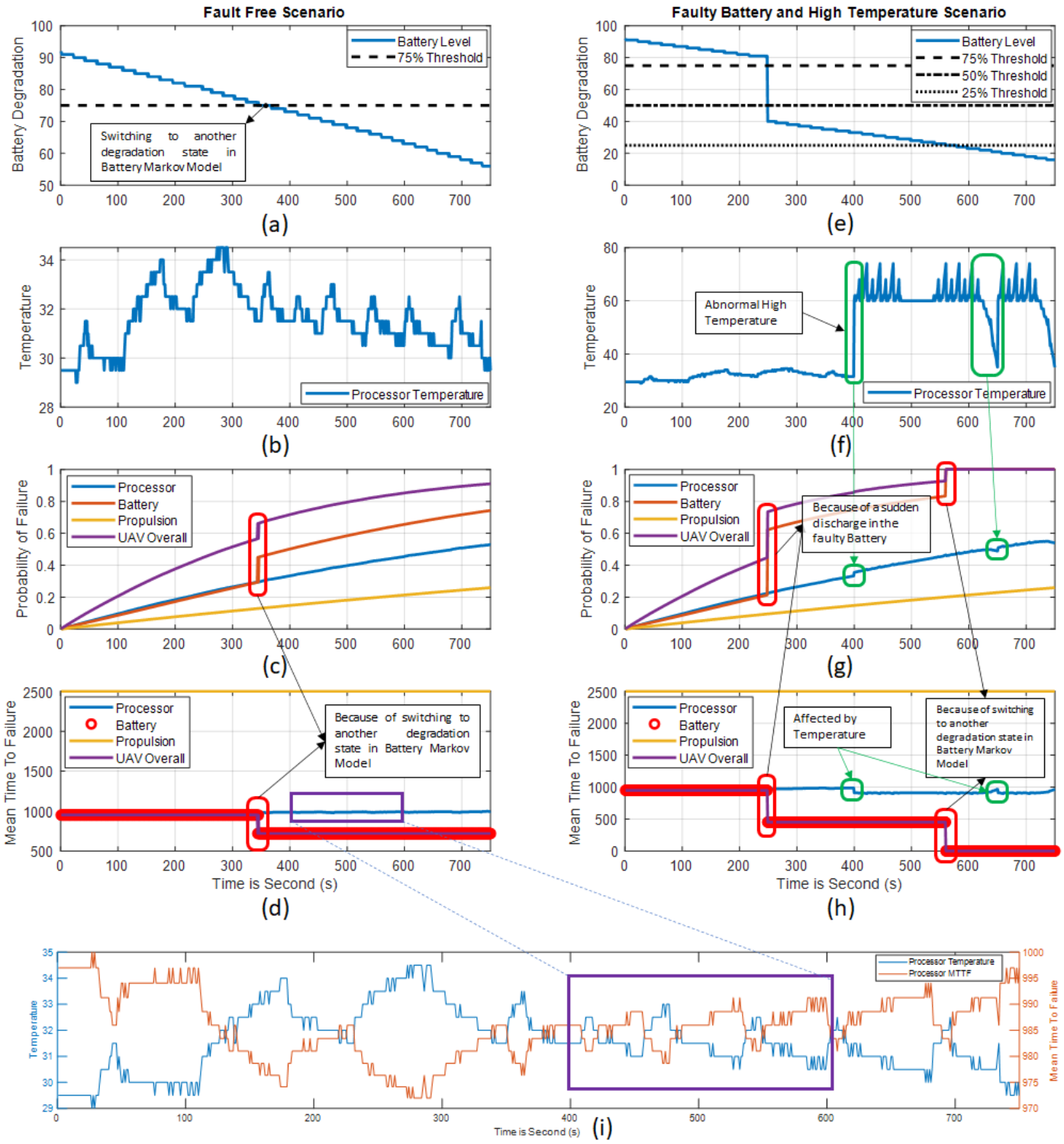


Figure 4-8. Fault-Free Scenario: (a) Battery degradation (battery level in percentage), (b) Processor Temperature (c) Probability of failure (d) Mean Time to failure -- Faulty Battery Scenario: (e) Battery degradation (battery level in percentage), (f) Processor Temperature (g) Probability of failure (h) Mean Time to failure -- (i) Processor's MTTF and temperature for the Fault Free Scenario

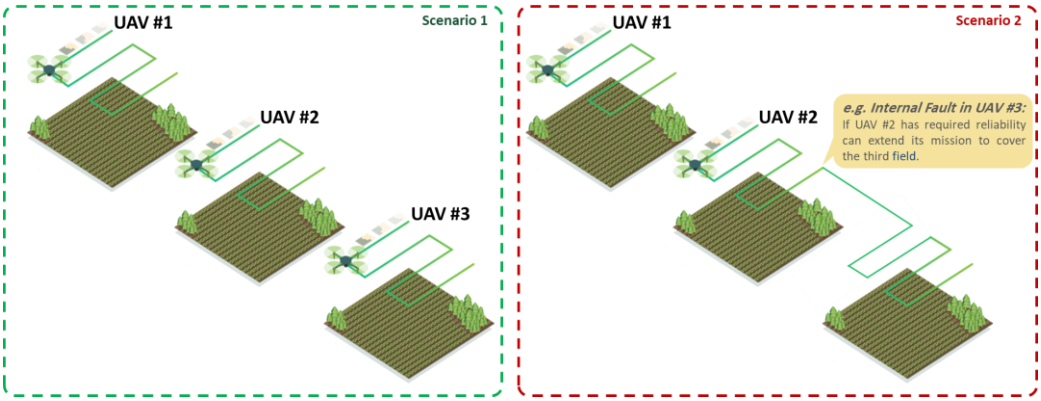


Figure 4-9. Application of SafeDrones for Multi-robot Precision Agriculture

Another example is provided in the following figure, considering the widely variable wind speeds encountered, a fault in the UAV that reduces controllability risks collision with a blade, causing damage to both UAV and wind turbine. The SafeDrones approach allows us to assess risk in real-time in response to the occurrence of faults by providing a runtime evaluation of reliability. Using this information, the drone can adapt its behaviour accordingly, e.g., by increasing safe distance to reduce risk of collision.

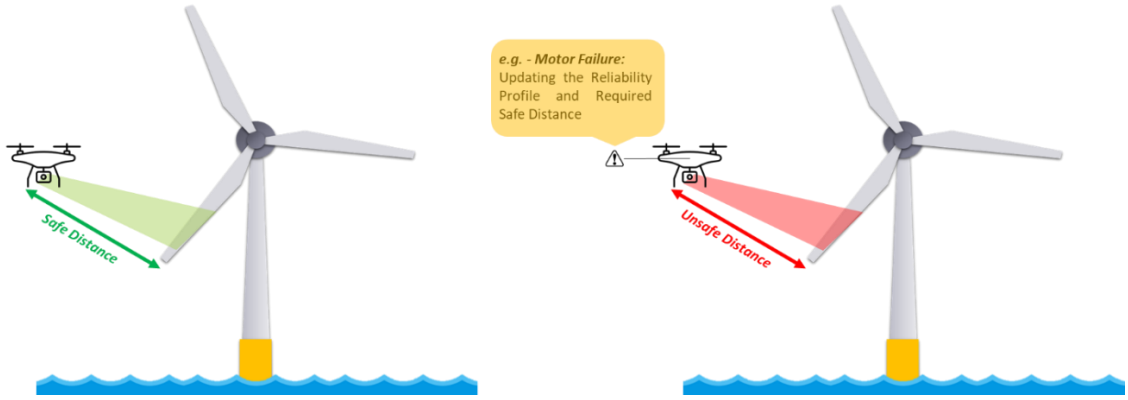


Figure 4-10. Application of SafeDrones for Offshore Wind Turbine Blade Inspection

4.7 Capabilities and Limitations of the Proposed Method

In this section, some of the capabilities and limitations of the proposed method are discussed. A number of suggestions are given to address limitations.

4.7.1 Capabilities of the Proposed Method

- This chapter discussed the idea of executable digital dependability identifier (EDDI) and provided an early prototype for it as a python package (SafeDrones). This combination of fault tree and Markov models is one of the first incarnations, and a novel form, of EDDI, which demonstrates the capabilities of the EDDI concept.

- A novel symptoms layer for the Dynamic Fault Tree has been proposed. It helps the model to be integrated with runtime monitoring.
- The case study shows how this form of EDDI can be used for power network inspection and how real-time reliability evaluation can be used to anticipate imminent failures and prevent accidents by recommending appropriate responses.

4.7.2 Limitations of the Proposed Method

- Currently, the provided experimental result is for a single robot and the multi-robot aspect is rather superficial. However, the proposed SafeDrones approach is capable of being considered for multi-UAV applications.
- Regarding the multi-UAV modelling, there is also a challenge to include the robot collaboration aspects. For instance, the communication between two robots and the effect of a communication loss in one robot can affect the reliability of the others. Moreover, the distance between two robots and the risk of calculation should be considered. These form areas of further work.
- The current proposed model is not capable of considering security events. However, integration is possible and pursued in work continuing this thesis in the SESAME project.
- The proposed approach is dependent on the accuracy and reliability of the monitoring and diagnosis unit. In this chapter, a way to include the uncertainty of symptoms has been discussed. However, for the scheme to work well, a reliable diagnosis and monitoring system is assumed.

4.8 Conclusion

To help address the problems of UAV reliability and risk assessment, particularly at runtime, where operational and environmental factors are hard to predict, the SafeDrones reliability modelling approach has been proposed by developing a new instantiation of the EDDI concept. The proposed model employs a combination of FTA with CBEs modelled as Markov Chains to support real-time reliability evaluation. As part of this, it introduces a novel symptoms layer to integrate with runtime monitoring data. To illustrate SafeDrones, we applied it to a power network inspection use case to show how real-time reliability evaluation can be used to anticipate imminent failures and prevent accidents by recommending appropriate responses.

Chapter 5

5 Addressing Distributional Shift with SafeML

This chapter focuses on the challenge and importance of AI safety and its directions. In particular, distributional shift is discussed a one of the main reasons for performance drop in Machine Learning algorithms. SafeML is proposed as a solution for safety monitoring of machine learning classifiers. The chapter provides several examples to show how SafeML can be effective for machine learning monitoring at runtime.

5.1 Artificial Intelligence and its Safety Challenges

Artificial Intelligence (AI) is growing fast in many applications. In parallel with this rapid growth, the concern of the society regarding the reliability, and safety assurance of those algorithms is raising. For safety-critical applications where human life or privacy, environment and economy are the topics in which the existing concern would be more serious. For instance, AI in medicine by (Wiens *et al.*, 2019), AI in autonomous systems

like self-driving cars by (Burton *et al.*, 2020; Du-Harpur *et al.*, 2020), AI in Military (Sharkey, 2019), and AI in economic by (Davenport *et al.*, 2019). In addition, different organizations and governmental institutes are trying to establish new rules, regulations, and standards for AI (Evans, 2020; *ISO/IEC JTC 1/SC 42: Artificial intelligence*, 2020).

The artificial intelligence and specifically machine learning can have harmful behaviour and put our life, privacy, environment, and money in risk (Mohseni *et al.*, 2023). It can be accidentally because of poor design, specifying the wrong objective function, having implementation error, choosing the wrong learning process, and using a poor or incomprehensive dataset for training (Tambon *et al.*, 2022). Thus, AI safety can be defined as a set of actions or endeavour to prevent any harm to humanity through AI. However, there are many perspectives and directions to be defined for AI Safety. In fact, (Amodei *et al.*, 2016) have addressed different research problems of certifying the modern machine learning systems operating in the field. They have categorized safety issues into five categories including A) Safe exploration, B) Robustness to distributional shift, C) Avoiding negative side effects, D) Avoiding "reward hacking" and "wire heading", E) Scalable oversight.

Statistical distance measure can be considered as a common method to measure distributional shift. Furthermore, in modern ML algorithms like Generative adversarial nets (GANs), different statistical distance or divergence is applied as a loss function such as Jensen-Shannon divergence (Goodfellow *et al.*, 2014), Wasserstein distance (Gulrajani *et al.*, 2017), and Cramer distance (Bellemare *et al.*, 2017). For dimension reduction, t-SNE (t-distributed stochastic neighbour embedding) algorithm uses the Kullback-Leibler divergence as a loss function (Laurens van der Maaten, 2014). In this report, the main focus will be on safety evaluation based on the statistical distance which is related to the robustness and distributional shift aspect of AI safety. A comprehensive study on dataset shift has been provided by (Quiñonero-Candela and Schwaighofer, 2009) and the dataset issues such as projection and projectability, simple and prior probability shift have been discussed. However, the study did not address the use of statistical distance and error bound to evaluate the dataset shift.

A resampling uncertainty estimation (RUE)-based algorithm has been proposed by (Schulam and Saria, 2019) to ensure the pointwise reliability of the regression when the test or field data set is different from the training dataset. The algorithm has created predictions ensemble through the modified gradient and Hessian functions for ML-based regression problems. The following table provides a summary of some relevant research

in the field. The table indicates whether an approach is model-specific (i.e., needs access to the model's underlying parameters and structure) or model-agnostic (i.e., treats the model as a black box and requires only the inputs and outputs), the type of input (e.g. images, tabular/numeric data), the type of task (regression or classification), and whether or not it can operate at runtime.

Table 5-1 provides a brief comparison between existing frameworks that aim to evaluate dependability attributes of ML/DL such as reliability and safety. The categorization includes brief explanation of their features, access type, model type, input type, task type, run-time evaluation capability and references. Each column of this table is explained as follows:

Name: This column provides the name(s) of each framework. It should be noted that some of them like SafeAI have different names for their sub-frameworks.

Features: The features column provides a brief introduction about the main focus of each framework and its main features.

Access Type: Two types of access including model-agnostic (MA) and model-specific (MS) access are considered. In the model-agnostic one the safety evaluation framework does not need to have access to the model's parameters or specification and just read its input-outputs. On the other hand, in the model-specific approaches, the framework needs to have access to the model's parameters and structure. Usually, the model specific approaches are tailored for deep neural network architectures and need to have access to the value of weights, biases, activation functions, number of layers, etc. While the model-agnostic approaches can cover wider range of existing approaches such as Gaussian processes, ensemble trees and support vector machines as well as deep neural networks. In addition, model-specific approaches may have a higher precision because of accessing the inner-parameters of a model. However, due to the intrusive nature of the approach, it cannot be used for private models, and it should be design for a specific model before evaluation. Furthermore, in some cases where a new deep neural network (e.g. deep evidential classification (Sensoy, Kaplan and Kandemir, 2018) or regression (Amini *et al.*, 2020) are used, the model-specific approaches may need to be re-designed and sometimes they need to come up with new solution to handle new models.

Model Type: The model type separated into two deep learning (DL) and machine learning (ML). As stated before, deep learning includes deep neutral network architectures such as feed forward neural networks, convolutional neural networks, and generative adversarial neural networks. The machine learning includes both deep neural

network architectures and other types of machine learning algorithms. In general, frameworks of ML type can support wider range of algorithms and those with DL are only restricted to deep neural network architectures.

Input Type: In this column, (T) stands for tabular input data, (I) stands for image input, (T) stands for text input, (G) stands for graph-based input, (TS) stands for time-series and (Q) stands for quantum-based input data. In SafeML, (G*) means, the framework has proposed a solution for graph-based input data, but it is not a stable version. From this column, it is clear that there is no available framework to handle quantum-based classifiers.

Task Type: Task type can be categorized into two classification and regression tasks. For graph-based input the classification task can be divided further into graph classification vs node classification. But it is not included as currently only SafeML proposed a solution for graph-based inputs, and it is designed for graph classification. In the future, such a table can be expanded further when there are more approaches supporting graph-based input.

Run-Time: For the real-time and safety critical systems, it vital to perform the evaluation at run-time. This column shows which framework provides a solution for run-time safety evaluation. When a framework has "No" in this column, it means the proposed evaluation is offline.

To have some examples from this table, (Zhao *et al.*, 2021) has proposed a framework called ReAsDL for reliability evaluation of machine learning algorithms. In this framework, as stated in feature column, the input space will be divided into small cells and the reliability of the ML/DL will be evaluated based on robustness and operational profile probability of those cells. The proposed approach is model agnostic, works with wide range ML/DL algorithms, currently only supports tabular input data, and it is designed for classification tasks. Furthermore, the proposed framework is good for offline reliability evaluation of machine learning classifiers and does not provide any run-time solution. (Aslansefat, Sorokos, *et al.*, 2020) has proposed a framework called SafeML for safety monitoring of machine learning algorithms. The first version of the framework was designed only for tabular input data but later has expanded for image in (Aslansefat *et al.*, 2021), the framework has also proposed a solution for graph-based inputs which is not stable yet.

Going through the following table, the framework proposed a solution that utilizes different statistical distance measures to quantify the input distributional shift and

estimate the model accuracy at run-time. The proposed solution is model-agnostic and can deal with wide range of ML and DL algorithms. SafeML has not offered any solution for regression tasks and can only be used for classification tasks. It should be noted that in this paper, SafeML will be used as a core function to evaluate the reliability of the machine learning classifiers.

Apart from comparing available frameworks for dependability evaluation of machine learning algorithms, let's focus on the available research works that have paved the road for this paper to establish a novel approach for reliability evaluation of machine learning classifiers. (Pietrantuono, Popov and Russo, 2020) has proposed an approach for the reliability evaluation of service-based software based on operational profile uncertainty. In this paper, it was assumed that the ground truth exists for each request. Thus, the approach was limited to offline model evaluation. Another point about this paper was that the author did focus on serviced-based software and the detail about how to form operational profile for black-box software and more specifically machine learning algorithms were missing. A year later (Zhao *et al.*, 2021) proposed a similar approach but specifically for machine learning classifiers. Zhao has focused on diving input data to cells to form the operational profile and did not study other possible scenarios to form the operational profile. So, in this paper not only different possible scenarios for forming the operation profile are studied but also a run-time solution is proposed.

Table 5-1. Summary of the existing ML dependability and safety approaches

Approach	**Features**	**Access Type**	**Input Type**	**Task Type**	**Run-time?**
DeepCert (Paterson *et al.*, 2021)	Aims to verify the robustness of DNN image classifiers in terms of sensitivity to image-based perturbations, e.g. blur, haze, contrast etc. Instead of measuring small pixel variations, these contextually relevant perturbations are encoded and quantified specifically. Demonstrated via integration with the Marabou DNN verification toolbox.	MS	I	C	No
DeepImportance (Gerasimou *et al.*, 2020)	Presents a systematic methodology for DL testing with new Importance-Driven Criteria. This allows a layer-wise functional understanding of DL components — the causal relationships between neurons — and thus makes it possible to assess the semantic diversity of a test set in terms of testing important neurons (in effect, a form of test coverage). Has an open-source tool.	MS	T/I	R/C	No

Approach	Features	Access Type	Input Type	Task Type	Run-time?
Marabou (Katz *et al.*, 2019)	A verification tool that can query fully connected and convolutional DNNs to provide a reachability and robustness assessment for a given neural network. Requires internal knowledge of the DNN to work as it performs a lazy search to locate solutions to non-linear constraints on the model.	MS	T/I	R/C	No
NN-Dependability (Cheng, Huang and Nührenberg, 2019)	Proposes new dependability metrics to measure the effect of uncertainty elimination in the ML/DL lifecycle. Also provides a formal reasoning engine to guarantee ML/DL behaviours.	MS	T/I TS	C	Yes
ReAsDL (Zhao *et al.*, 2021)	Focuses on the impact of the operational profile on robustness. Divides the input space into small cells and evaluates the reliability of the ML/DL based on robustness and operational profile of those cells. Prototype tool available online.	MA	T	C	No
Safe AI (Gehr *et al.*, 2018)	A collection of related approaches, e.g. DiffAI, DL2, AI^2, PRIMA etc. Their main focus is on possible perturbation to the input space (adversarial examples) and providing robust, safe, and interpretable solutions and certifications.	MS	T/I	C	No
SAFE-DNN (She *et al.*, 2020)	Investigates the property inference in DNNs as part of the verification process. Combines supervised and unsupervised learning by augmenting the feature space of the (supervised) DNN with features extracted by an (unsupervised) spiked neural network, increasing robustness of the DNN.	MS	T/I	C	No
Safeguard AI (Lee *et al.*, 2017)	Calculates probability for out-of-distribution input as confidence loss and adds that probability to the existing loss function. Intended for use during training by identifying OOD samples and generating improved training data using a GAN to minimise confidence loss.	MS	I	C	No
SafeML (Aslansefat, Sorokos, *et al.*, 2020; Aslansefat *et al.*, 2021)	Uses statistical distance measures to quantify the distributional shift. Then estimates the accuracy, updates the uncertainty, and evaluates reliability.	MA	T/I TS/G	C	Yes

Approach	Features	Access Type	Input Type	Task Type	Run-time?
(Kläs and Sembach, 2019; Kläs and Jöckel, 2020)Wrappers	Focuses on three main ML verification domains: model performance, input quality, and scope compliance. Provides a set of useful functions to evaluate the existing uncertainties in each step.	MA	T/I	C	No

Upon researching the literature, there is no conducted research work so far in which safety monitoring and accuracy estimation of the ML-based classification be addressed using ECDF-based statistical distance measures. In this thesis, a modified version of statistical distance is used to compare the dataset during the training procedure and the field test. Then a novel human-in-loop procedure has been proposed to estimate and certify the accuracy of the system in different scenarios. Different examples have been provided to show the capabilities and limitations of the proposed approach.

The rest of this chapter is organised as follows: In section II, the problem definition is provided. The proposed method is addressed in section III. Numerical results are demonstrated in section IV with a brief discussion. The capabilities and limitations of the proposed method are summarised in section V and the thesis terminates with a conclusion.

5.2 Problem Definition

The classification algorithms have different applications. For instance, abnormality detection can be one of them. A simple classifier can be a line or threshold. Consider a hypothetical measurement (e.g. Temperature) as shown in Figure 5-1 and can be defined as follows:

$$D(t) = \begin{cases} Class\ 1 & 0 \leq t \leq 100 \\ Class\ 2 & 101 \leq t \leq 200 \end{cases} \quad (5\text{-}1)$$

The measurement $D(t)$ has included two classes: “Class 1” and “Class 2”. For instance, they can represent the normal and abnormal state of a system. From time 0 to 100 is class 1 and from 101 to 200 is class 2.

Figure 5-1. A hypothetical measurement (i.e. from 0 to 100 is Class 1 and from 101 to 200 is Class 2).

The probability density function of the measurement can be estimated as shown in Figure 5-2. In this figure, the threshold has been represented with a red vertical dash-line and value of four. The area with an overlap in this figure can cause missed and false detection (also called false positive/type I error and false-negative/type II error).

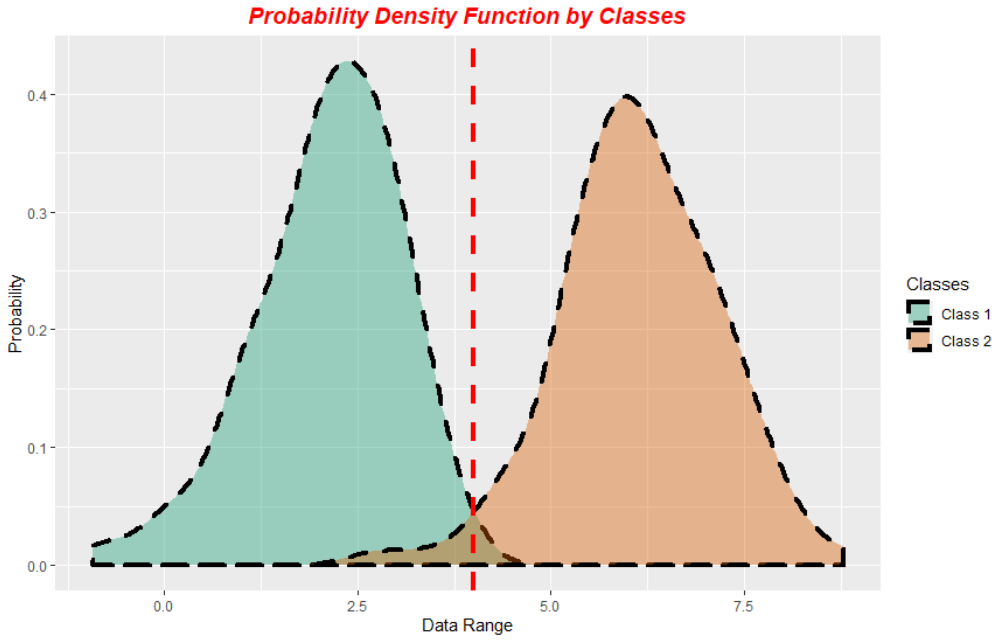


Figure 5-2. The estimated probability density function for both Class 1 and Class 2 with a threshold equal to four.

Looking at the figure of probability density functions, the area that two probability density functions are merging can cause the miss-classification and consequently the error. Probability of the error or miss-classification can be calculated using the following equation (Theodoridis and Koutroumbas, 2009). It should be noted that the error

probability is also related to the threshold (x considered as threshold value). For more details check (Aslansefat, Bahar Gogani, *et al.*, 2020).

$$\mathrm{P}(error) = \int_{-\infty}^{+\infty} P(error|x)P(x)dx \tag{5-2}$$

In which the $\mathrm{P}\vdash(error|x\backslash right)$ can be calculated through calculating the minimum of both probability density functions as the following equation. The minimization is subject to variation of threshold value from -inf to +inf.

$$\mathrm{P}(error|x) = \min[P(Class\ 1|x), P(Class\ 2|x)] \tag{5-3}$$

By dividing the space into two regions as R_1 and R_2, the probability of error can be written with parts.

$$\begin{aligned} P(error) &= P(x \in R_1, Class\ 1) + P(x \in R_2, Class\ 2) \\ &= \int_{R_1} P(x|Class\ 1)P(Class\ 1)\, dx \\ &+ \int_{R_2} P(x|Class\ 2)P(Class\ 2)\, dx \end{aligned} \tag{5-4}$$

To ease the minimization problem, consider the following inequity (Fukunaga, 1992).

$$min[a, b] \leq a^{\lambda} b^{1-\lambda}\ where\ a, b \geq 0\ and\ 0 \leq \alpha \leq 1 \tag{5-5}$$

Equation (5-3) can be rewritten as follows:

$$\begin{aligned} P(error|x) &= min[P(Class\ 1|x), P(Class\ 2|x)] \\ &= min\left[\frac{P(x|Class\ 1)P(Class\ 1)}{P(x)}, \frac{P(x|Class\ 2)P(Class\ 2)}{P(x)}\right] \end{aligned} \tag{5-6}$$

Using the inequity rule and equation above, the conditional probability of error can be derived as:

$$\begin{aligned} &P(error|x) \\ &\leq \left(\frac{P(x|Class\ 1)P(Class\ 1)}{P(x)}\right)^{\lambda} \left(\frac{P(x|Class\ 2)P(Class\ 2)}{P(x)}\right)^{1-\lambda} \end{aligned} \tag{5-7}$$

The following equation can be obtained using equations (5-2) and (5-7).

$$\begin{aligned} &P(error) \\ &\leq \big(P(Class\ 1)\big)^{\lambda}\big(P(Class\ 2)\big)^{1-\lambda} \int_{-\infty}^{+\infty} \big(P(x|Class\ 1)\big)^{\lambda}\big(P(x|Class\ 2)\big)^{1-} \end{aligned} \tag{5-8)}$$

It safety assurance, it is so important to consider the worth case scenario which can lead us to the following equation known as Chernoff upper bound of error (Fukunaga, 1992).

$$P(error) = \left(P(Class\ 1)\right)^{\lambda}\left(P(Class\ 2)\right)^{1-\lambda}\int_{-\infty}^{+\infty}\left(P(x|Class\ 1)\right)^{\lambda}\left(P(x|Class\ 2)\right)^{1-} \quad (5\text{-}9)$$

If the probability distributions of the features obey normal or exponential distribution families, the integral part of above equation can be solved through the following equation (Fukunaga, 1992).

$$\int_{-\infty}^{+\infty}\left(P(x|Class\ 1)\right)^{\lambda}\left(P(x|Class\ 2)\right)^{1-\lambda}dx = e^{-\theta(\lambda)} \quad (5\text{-}10)$$

The $\theta(\lambda)$ can be calculated using (5-11) where μ and Σ are mean vector and variance matrix of each class respectively.

$$\theta(\lambda) = \frac{\lambda(1-\lambda)}{2}[\mu_2-\mu_1]^T[\lambda\Sigma_1+(1-\lambda)\Sigma_2]^{-1}[\mu_2-\mu_1] + 0.5\ log\frac{|\lambda\Sigma_1+(1-\lambda)\Sigma_2|}{|\Sigma_1|^{\lambda}|\Sigma_2|^{(1-\lambda)}} \quad (5\text{-}11)$$

If you consider $\alpha = 0.5$ the above equation become the Bhattacharyya distance which can be proven that this value is the optimal value when $\Sigma_1 = \Sigma_2$ (Fukunaga, 1992; Nielsen, 2014). In this study, the Bhattacharyya distance will be used to demonstrate the idea. It should be noted that is some case the calculated error bound might be higher than the real value. However, it is accepted because in Safety evaluation the worth case scenario is needed to be considered. As the P$(error)$ and P$(correct)$ are complimentary. Then the probability of having correct decision can be calculated using the following equation:

$$P(correct) = 1 - \sqrt{P(Class\ 1)P(Class\ 2)}\,e^{-\theta(\lambda)} \quad (5\text{-}12)$$

The Chernoff upper bound of error is usually used as a measure of separability of two classes of data, but with the above equation measures the similarity of two classes. In other words, in an ideal situation if you calculate the P$(error)$ of a class, with itself, the response should be equal to one while P$(correct)$ should be zero. The idea is to show whether data distribution in the training procedure is the same as the data distribution in the field or not. The following assumptions are needed to be considered for this method:

- The covariance matrix of the data should be positive-definite matrix to have determinant and be invisible. Therefore, in each dataset, columns with all zero, Nan or Inf values should be removed. Also, columns with zero variance should be removed or it can be possible to assign a small number like 1e-6 as their variance.
- Non-coherent datasets (e.g., XOR datasets) can reduce the accuracy of the algorithm.
- For datasets like Circular and Spiral one, it is suggested to convert the data to the polar coordination before the analysis.

Note: Using dimension reduction algorithms such as PCA and t-SNE can solve the existing limitations of the proposed algorithm.

There is a relation between existing statistical distance measures as illustrated in Figure 5-3. So, it will be possible to extend the error bound probability calculations for other distance or divergence measures (Nielsen, 2018). However, computing the error bound using some distance or divergence measure algorithms like Kullback-Leibler divergence would be computationally complex (Zahm *et al.*, 2018).

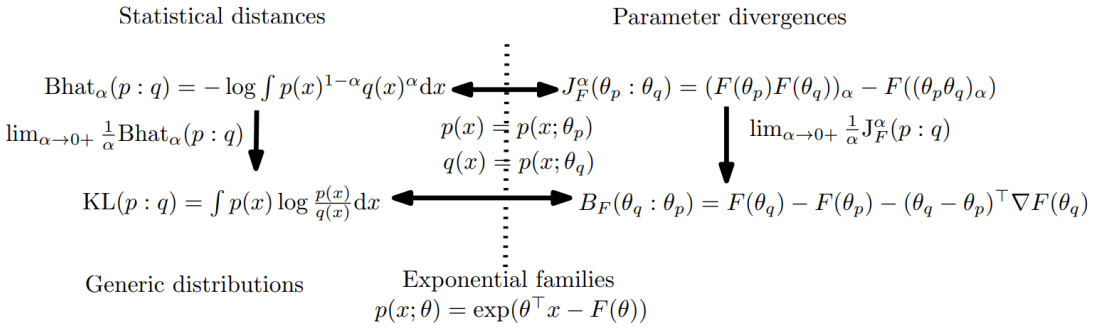


Figure 5-3. The relation between statistical distances like Bhattacharyya and Kullback-Leibler and parameter divergences like Jensen divergence and Burbea-Rao distances (Nielsen, 2018)

Considering $\mathrm{P}(Class\ 1) = \mathrm{P}(Class\ 2)$, and converting the integral part to the cumulative distribution function as follows:

$$P(error) = \left(\int_{R_1} P_{Class\ 1}(x)\, dx + \int_{R_2} P_{Class\ 2}(x)\, dx \right) = \Big(\big(1 - F_{Class\ 1}(x)\big) + \big(F_{Class\ 2}(x) - 0\big) \Big) \tag{5-13}$$

Using Kolmogorov-Smirnov distance, probability of error upper bound can be achieved as the following equation:

$$P(error) \approx \sup_x \big(F_{Class\ 2}(x) - F_{Class\ 1}(x) \big) \tag{5-14}$$

As future research, other novel statistical distance measures can be used for upper bound error probability estimation. For example, (Hadjeres and Nielsen, 2020) have

proposed a novel Schoenberg-Rao distances that used both Entropy-based and geometry-aware statistical Hilbert distances.

5.3 SafeML Approach

Figure 5-4 illustrates the flowchart of the proposed approach. In this flowchart, there are two main sections including training phase and application phase. A) The training phase is an offline procedure in which a trusted dataset will be used to train the intelligent algorithm that can be a machine learning or deep learning algorithm. This study will focus on the classification ability of machine learning. Thus, using a trusted dataset the classifier will be trained and its performance will be measured with existing KPIs.

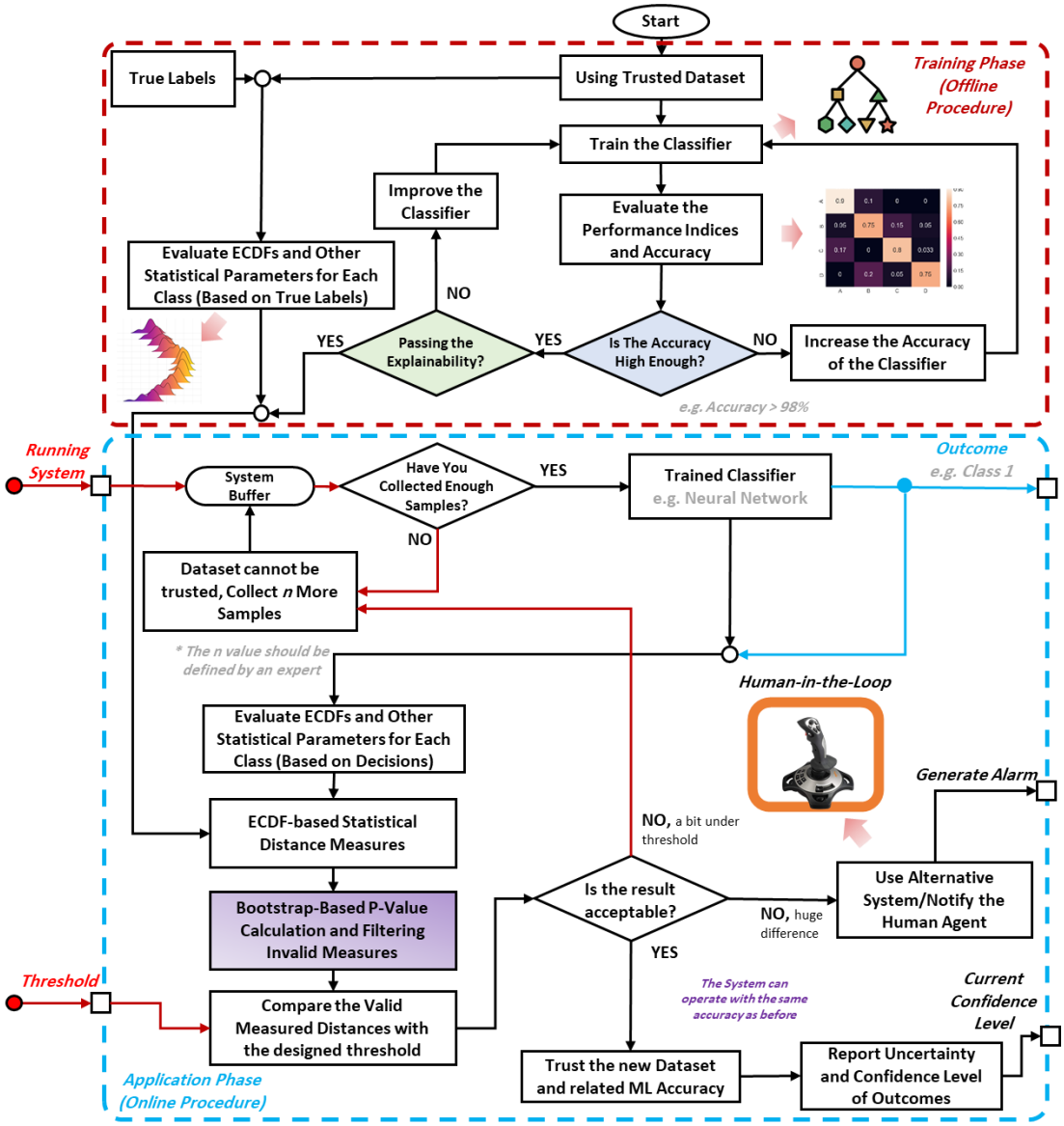


Figure 5-4. Flowchart of the proposed approach (Aslansefat *et al.*, 2021)

Meanwhile, the probability density function and statistical parameters of each class will be estimated and stored to be used for comparison. B) The second phase or

application phase is an online procedure in which real-time and unlabelled data is going to be feed to the system. For example, consider an autonomous car has been trained to detect obstacles and it should prevent a collision. Therefore, in the application phase, the trained classifier should distinguish between the road and other objects. One important and critical issue in the application phase is that the data does not have any label. So, it cannot be assured that the classifier can operate as accurate as of the training phase. In the application phase, the untrusted labels of the classifier will be used and similarly, the probability density function and statistical parameters of each class will be extracted. Using modified Chernoff Error bound the statistical difference of each class in the training phase and application phase is compared. If the statistical difference was very low, the classifier results and accuracy can be trusted (In this example the autonomous car continuous its operation), if the statistical difference was low, the system can ask for more data and re-evaluation to make sure about the distance. In case of having more statistical difference, the classifier results and accuracy is no longer valid, and the system should use alternative approach or notify human agent (In this example, the system will ask the driver to take the control of the car).

5.4 ECDF-based Statistical Distance Measures

The main core of the SafeML approach is the statistical distance measures. In this section, five well-known distance measures including Kolmogorov-Smirnov, Kuiper, Anderson-Darling, Cramer-Von-Mises, and Wasserstein will be addressed briefly. The following equation demonstrates the Kolmogorov-Smirnov Distance (KSD) for two univariate distributions of $F_1(x)$ and $F_2(x)$where $F_1(x)$ can be the ECDF of the train dataset and $F_2(x)$can be the ECDF of test dataset with respect to their class labels. For multi-variate dataset the KSD can be calculated for each feature and finally, the aggregated value can be calculated for the whole dataset.

$$KSD \approx \sup_{x \in R}\big(F_1(x) - F_2(x)\big) \tag{5-15}$$

Similar to KSD, the Kuiper Distance (KD) can be calculated using the following equation. The KD is more efficient when there is a situation in which two univariate datasets have the same mean value and different variance like circle and spiral datasets.

$$\mathrm{KD} \approx \max_{x \in R}\big(\mathrm{F}_1(x) - \mathrm{F}_2(x)\big) + \min_{x \in R}\big(\mathrm{F}_1(x) - \mathrm{F}_2(x)\big) \tag{5-16}$$

The Cramer-Von-Mises Distance (CVMD) uses summation like (5-17) and instead of max in KSD and rely on the step-size and sample rate of the features.

$$\mathrm{CVMD} \approx \sum_{x \in X} |\mathrm{F}_1\,(x) - \mathrm{F}_2\,(x)| \tag{5-17}$$

If one can form a joint distribution between two ECDF and divide the CVMD values by joint distribution values then the Anderson-Darling Distance (ADD) can be obtained as (5-18).

$$ADD \approx \sum_{x \in X} \frac{|F_1(x) - F_2(x)|}{D(x)\big(1 - D(x)\big)} \tag{5-18}$$

The Wasserstein Distance (WD) can be defined as (5-19). The WD is calculating the distance between two ECDFs and its functionality is similar to CVMD.

$$WD \approx \int_{-\infty}^{+\infty} |F_1(x) - F_2(x)| dx \tag{5-19}$$

To answer the question that why other statistical distance measures such as t-test, Chernoff distance, Kullback Leibler divergence, etc. were not addressed in this paper, It should be noted that most of the these methods rely on mean and variance values of the distributions and they are less sensitive to the geometry of the distribution in comparison to the ECDF-based distances. Their implementations are available on the SafeML GitHub repository for interested readers.

One of the main drawbacks in the first version of SafeML (Aslansefat, Sorokos, *et al.*, 2020) was that all the distance measures were accepted in the algorithm. While considering the p-values, some of the measures cannot be accepted and should be removed from the procedure. To calculate the p-values in the aforementioned algorithms, the Bootstrap algorithm is used as shown in the following Algorithm (Gilleland, 2020). This algorithm can be performed for all aforementioned ECDF-based distance measures. The Algorithm is considered for the WD and can be expanded for the other distances with similar structure. The aim of the algorithm is to provide both WD and the p-value associated with the outcomes. For a univariate example, X and Y are the inputs and the Bootstrap algorithm will be executed for 1e5 iterations. The XY will be the concatenated version of X and Y and in each iteration two random samples will be extracted from XY. The WD of X and Y will be compared with the WD of two random samples named boostWD and every time that boostWD > WD, the `bigger' counter will increase. Finally, after performing the 1e5 iterations, the p-value named as pVal will be calculated as shown in this algorithm.

Algorithm 5-1

```
MaxItr = 1e5;
 WD = Wasserstein_Dist(X,Y);
 XY = Concatenate(X,Y);
 LX = len(X);  LY = len(Y);
 n = LX + LY;
 For ii in 1 to MaxItr:
       e = random.sample(range(n), LX);
       f = random.sample(range(n), LY);
       boostWD = Wasserstein_Dist(XY[e],XY[f]);
         If{boostWD > WD
               bigger = 1 + bigger;
pVal = bigger/MaxItr;
Return pVal, WD;
```

5.5 Results

In this section, the proposed method is applied on I) typical synthetic benchmarks, II) on security intrusion dataset, and III) traffic sign recognition benchmark for ML classification.

The proposed method has been implemented in three different programming languages including R, Python and MATLAB. Regarding R programming, three well-known benchmarks have been selected: a) the XOR dataset, b) the Spiral dataset and c) the Circle dataset. Each dataset has two features (i.e. input variables) and two classes. The following figure illustrates the scatter plots of the selected benchmarks. More examples and benchmarks are available at https://github.com/ISorokos/SafeML.

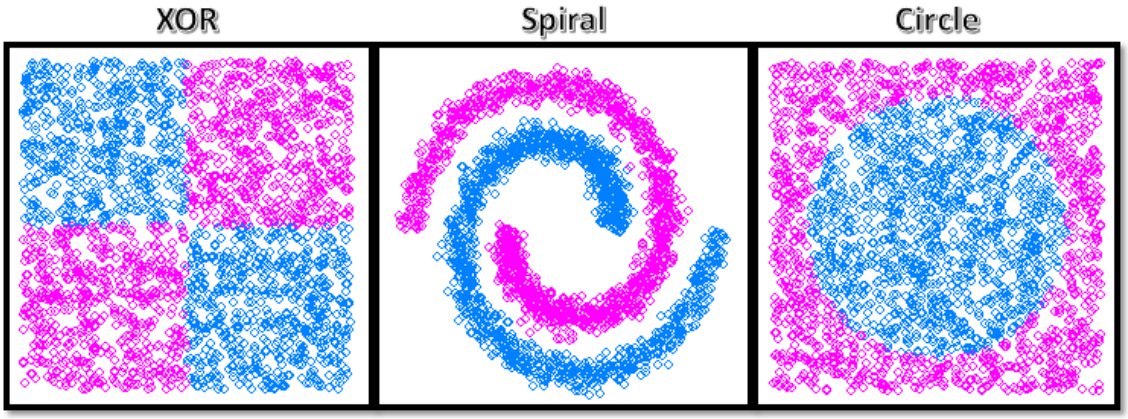


Figure 5-5. Scatter plot of XOR, Spiral and Circle Benchmarks

5.5.1 Example 1. XOR, Spiral and Circle Benchmarks

To start the ML-based classification, 80 percent of each dataset was used for training and testing and 20 percent of the dataset has been used for validation, with 10-fold cross-validation. Both linear and nonlinear classifiers have been selected for classification. The Linear discriminant analysis (LDA) and the Classification And

Regression Tree (CART) are used as linear methods. Moreover, The Random Forest (RF), K-Nearest Neighbours (KNN) and Support Vector Machine (SVM) are applied as nonlinear methods. As KPIs, the accuracy and Kappa measure are used to measure the performance of each classifier. Finally, as Empirical Cumulative Distribution Function (ECDF)-based statistical distance measures, the Kolmogorov-Smirnov Distance (KSD), Kuiper Distance, Anderson-Darling Distance (ADD), Wasserstein Distance (WD), and a combination of ADD and Wasserstein-Anderson-Darling Distance (WAD) have been selected for evaluation.

XOR Dataset: The XOR dataset has two features and two classes in which features have the same mean and variance characteristics. The following table compares the estimated accuracy based on the ECDF measures with the Minimum True Accuracy (MTA) and the Average True Accuracy (ATA) over 10 folds. For instance, the second column of this table provides the estimated accuracy based on the KSD measure. As a matter of safety, MTA is more important because it represents the worst-case scenarios, where the lowest accuracy may be experienced and impact safety. We observe that the KSD measure reports low accuracy for the LDA classifier (~.77). Instead, the ADD and WAD measures significantly overestimate the accuracy of the LDA.

Table 5-2. Comparison of estimated accuracies vs minimum true accuracy for XOR dataset

Method	KSD	Kuiper	ADD	WD	DTS	BD	MTA	ATA
LDA	0.7722165	0.7706	0.902818	0.755064	0.985666	0.154506	0.508333	0.591211
CART	0.9281788	0.921982	0.987722	0.925458	0.995211	0.497243	0.987448	0.994158
KNN	0.9305751	0.913063	0.993151	0.958768	0.997076	0.497102	0.974895	0.986665
SVM	0.9310446	0.917586	0.993489	0.958191	0.997064	0.496731	0.979167	0.987917
RF	0.9296264	0.910749	0.992742	0.957821	0.997018	0.496856	0.995833	0.998333

Based on Table 5-2, Table 5-3 represents the (absolute) difference between accuracy estimations of each measure and the MTA of each classifier. The ADD, WD and WAD measures have the best accuracy estimations overall. In particular, when a LDA classifier is used, the WD measure provides an estimated accuracy with comparatively less error.

Table 5-3. Difference between Distance Measures and MTA for XOR dataset

Method	KSD	Kuiper	ADD	WD	DTS	BD
LDA	0.2638832	0.262267	0.394484	0.246731	0.477333	0.353828
CART	0.0592689	0.065466	0.000274	0.06199	0.007763	0.490205
KNN	0.0443203	0.061833	0.018256	0.016127	0.02218	0.477793
SVM	0.0481221	0.06158	0.014322	0.020976	0.017897	0.48231
RF	0.0662069	0.085084	0.003092	0.038012	0.001184	0.499102

Spiral Dataset: Similar to the XOR dataset, the proposed method can be applied for the spiral dataset. Table 5-4 presents difference between ECDF-based distance measures and minimum true accuracy for this dataset. For brevity, for this dataset and the next one, only the difference table is provided. Based on this table, the KSD and Kuiper distance have better estimation for accuracy of the classifiers for the spiral dataset.

Table 5-4. Difference between Distance Measures and MTA for Spiral dataset

Method	KSD	Kuiper	ADD	WD	DTS	BD
LDA	0.099447	0.088252	0.269975	0.248396	0.528852	0.043445
CART	0.056131	0.031092	0.149191	0.09477	0.158529	0.355675
KNN	0.047526	0.075598	0.001468	0.014756	0.002734	0.496559
SVM	0.047526	0.075598	0.001468	0.014756	0.002734	0.496608
RF	0.024471	0.050261	0.018778	0.003885	0.019643	0.479893

Circle dataset: The circle dataset has similar statistical characteristics with the spiral dataset. Table 5-5 provides the difference between ECDF-based distance measures and MTA for this dataset. As can be seen, the worst accuracy estimation is related to the accuracy estimation of the LDA classifier. For the LDA, the Kuiper distance estimates with less error, with the KSD and WD being in second and third place respectively.

Table 5-5. Difference between Distance Measures and MTA for Circle dataset

Method	KSD	Kuiper	ADD	WD	WAD	BD
LDA	0.329391	0.250345	0.412382	0.34745	0.498267	0.23667
CART	0.114312	0.019111	0.168596	0.099549	0.243227	0.455675
KNN	0.004833	0.037554	0.027649	0.010871	0.02775	0.498459
SVM	0.016133	0.043604	0.019147	0.001695	0.01935	0.498808
RF	0.004663	0.034529	0.027776	0.012814	0.02782	0.468893

5.5.2 Example 2: Security Intrusion Dataset

This case-study applies the proposed method towards the CICIDS2017 dataset, which was originally produced by (Sharafaldin, Lashkari and Ghorbani, 2018) at the Canadian Institute for Cyber Security (CICS) as an aide to the development and research of anomaly-based intrusion detection techniques for use in Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) (Panigrahi and Borah, 2018).

The labelled dataset includes both benign (Monday) and malicious (Tuesday, Wednesday, Thursday, Friday) activity. The benign network traffic is simulated by abstraction of typical user activity using a number of common protocols such as HTTP, HTTPS, FTP and SHH. Benign and malicious network activity is included as packet payloads in packet capture format (PCAPS).

Wednesday Attack: This attack occurred on Wednesday, July 5, 2017, and different types of attacks on the availability of the victim's system have been recorded, such as DoS / DDoS, DoS slow loris (9:47 – 10:10 a.m.), DoS Slow http test (10:14 – 10:35 a.m.), DoS Hulk (10:43 – 11 a.m.), and DoS Golden Eye (11:10 – 11:23 a.m.). Regarding the cross-validation, a hold-out approach has been used, in which 70 percent of data has been randomly extracted for testing and training and the rest has been used for accuracy estimation. Additionally, traditional classifiers including 'Naive Bayes','Discriminant Analysis','Classification Tree', and 'Nearest Neighbor' have been used. Figure 5-6 shows the confusion matrix when Naive Bayes classifier is used.

Figure 5-7 shows a sample result of six statistical measures (one PDF-based and five ECDF-based) vs. the classifier's accuracy. It is expected to have low distance (from the ECDF measures) when the accuracy is high and vice versa. As can be seen in this figure, the Kuiper distance measure performs better. However, these results were derived from only one iteration of the random hold-out process. Therefore, the process should be iterated over several times and the performance of each distance measure on average should be reviewed.

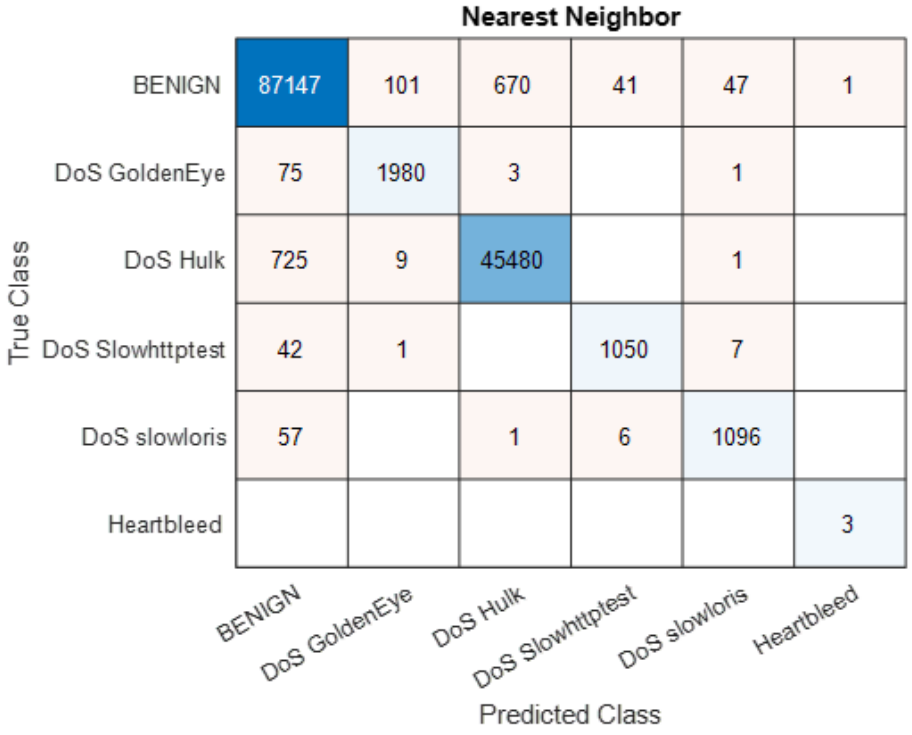


Figure 5-6. Confusion matrix for Wednesday Security Intrusion Detection in CICIDS2017 Dataset

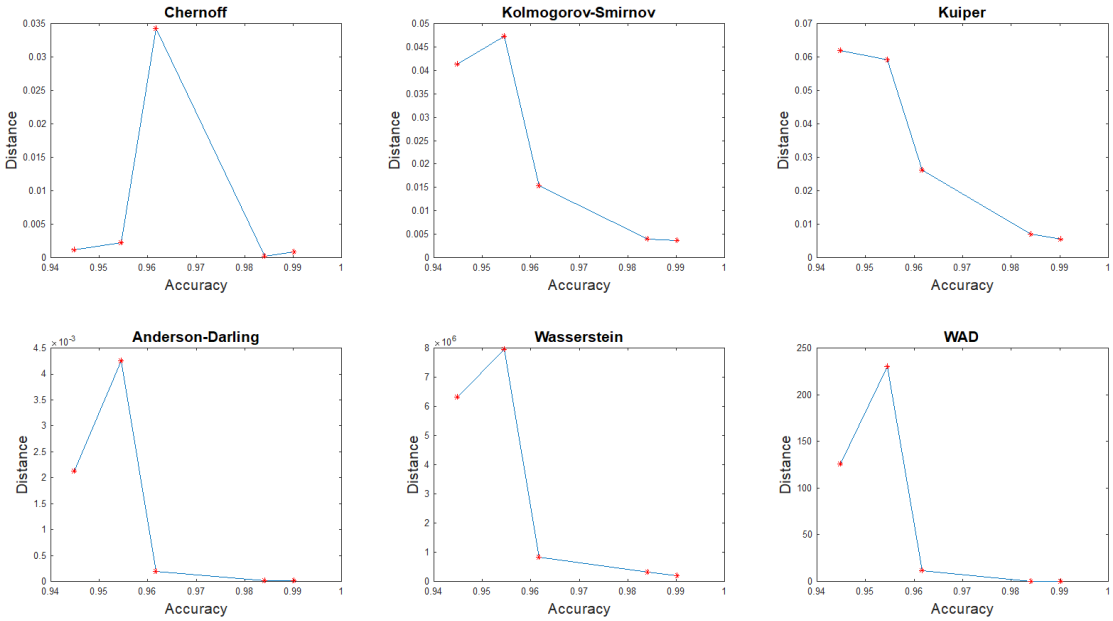


Figure 5-7. Sample statistical distance measures vs. accuracy

Figure 5-8 has been generated over 100 iterations. For each iteration, 70 percent of the data has been randomly extracted for testing and training and the rest has been used for accuracy estimation. The figure shows the box plot of the statistical distance measurements vs. the evaluated accuracy over 100 iterations. By observing the average values (red lines) of each box plot, the relationship between each measure and the average change in accuracy can be understood. In addition, this plot shows which method has less variation. For instance, the Kuiper distance and WD have the best performance while Chernoff has the least performance.

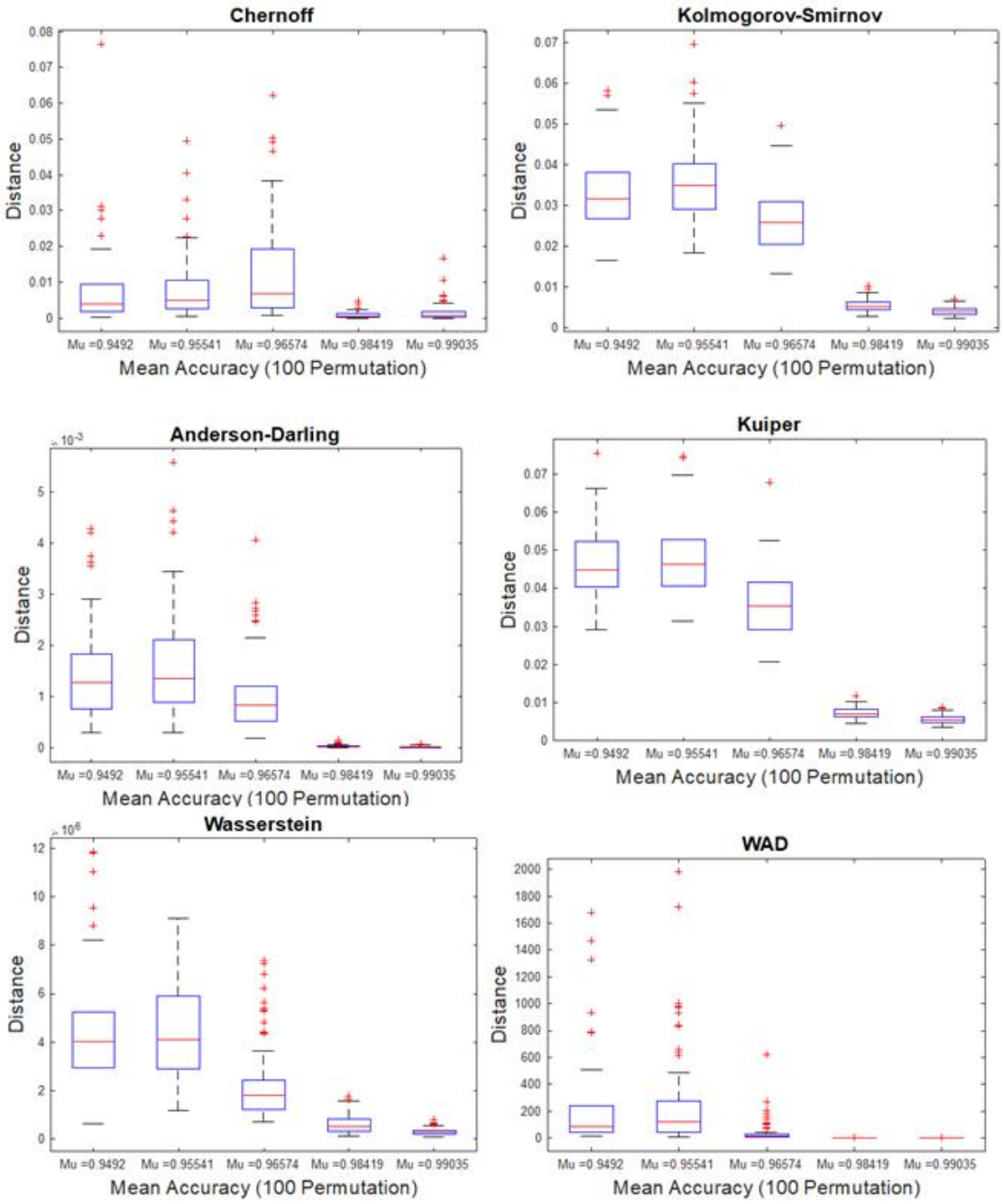


Figure 5-8. Box plot of statistical distance measures vs. accuracy over 100 iterations

Thursday Attack: This attack occurred on Thursday, July 6, 2017, and various attacks, such as the Web Attack – Brute Force (9:20 – 10 a.m.), Web Attack – XSS (10:15 – 10:35 a.m.), and Web Attack – SQL Injection (10:40 – 10:42 a.m.) have been recorded. Figure \ref{fig_Th01} shows the confusion matrix for Thursday morning's security intrusion in the CICIDS2017 dataset when the Naive Bayes classifier is applied. Similar to Wednesday, 70 percent hold-out cross validation is used for this dataset. As can be

seen, this dataset has four classes, and the classifier has problem to detect the last class or last type of intrusion.

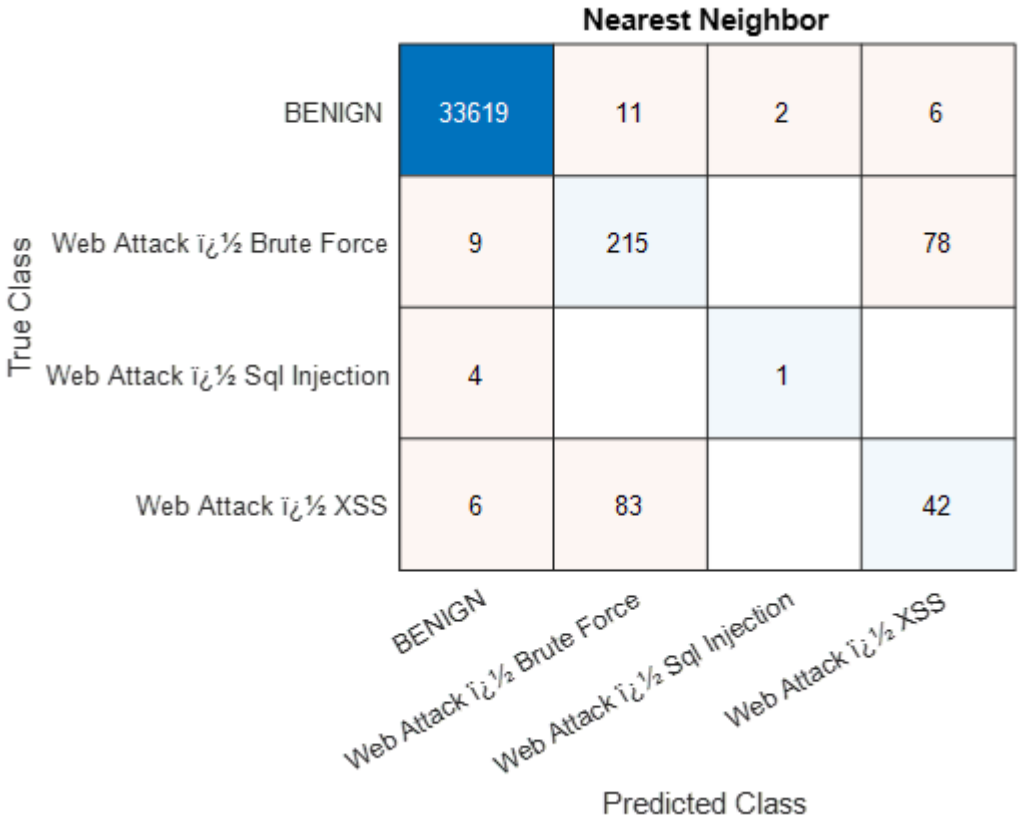


Figure 5-9. Confusion matrix for Thursday Security Intrusion Detection in CICIDS2017 dataset

The following figure shows a sample result of six statistical measures (Chern-off and five ECDF-based measures) vs. accuracy of the classifier. In this sample, the Kolmogorov-Smirnov and Kuiper measures have better performance.

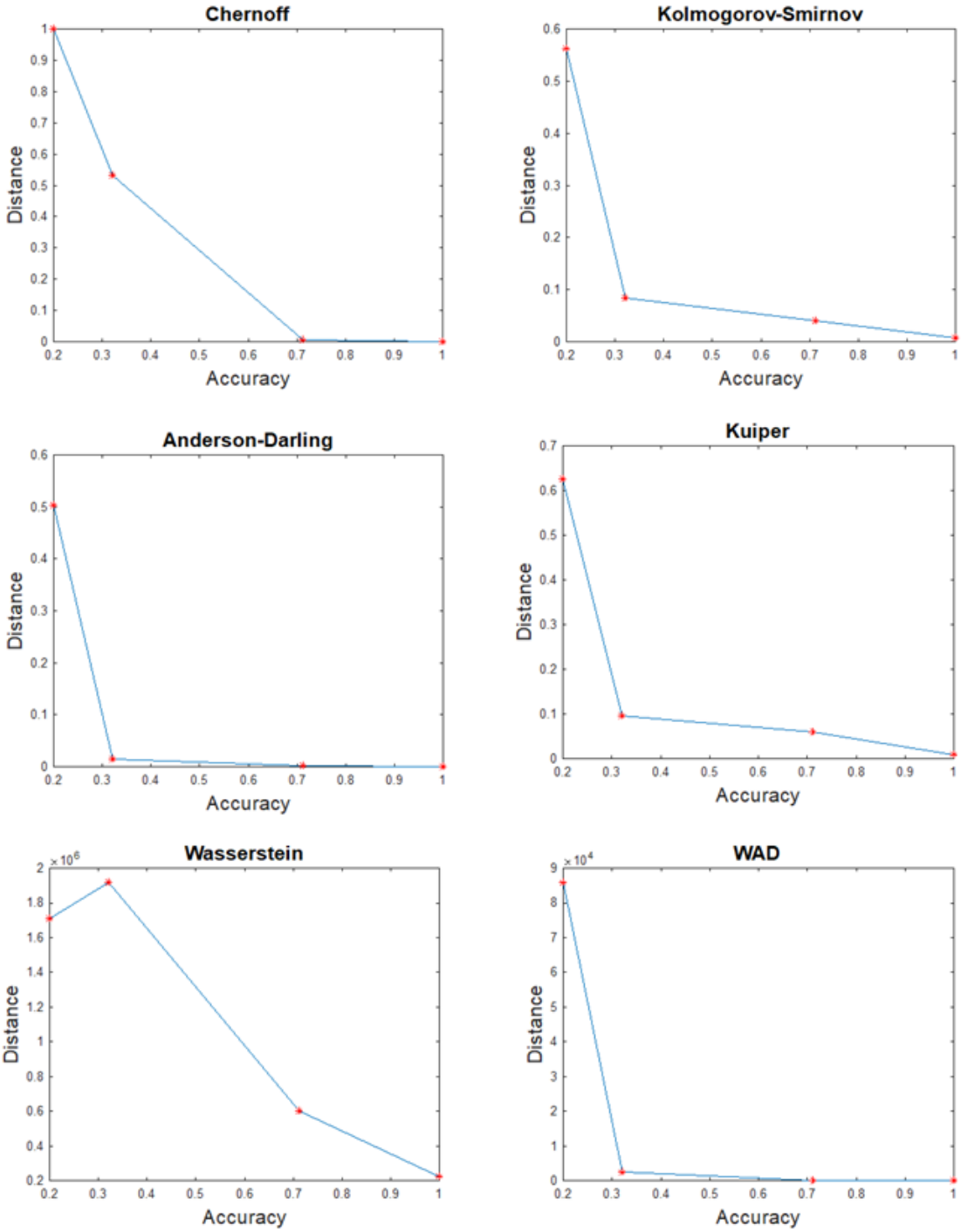


Figure 5-10. Sample statistical distance measures vs. accuracy for Thursday Security Intrusion Detection in CICIDS2017 dataset

Similar to the previous example, the following figure has been generated over 100 times and the box plot of Figure \ref{fig_Th03} can be seen. In this figure, the Kolmogorov-Smirnov, Kuiper, and Wasserstein distance measures have a better performance, however, their decision variance is a bit high.

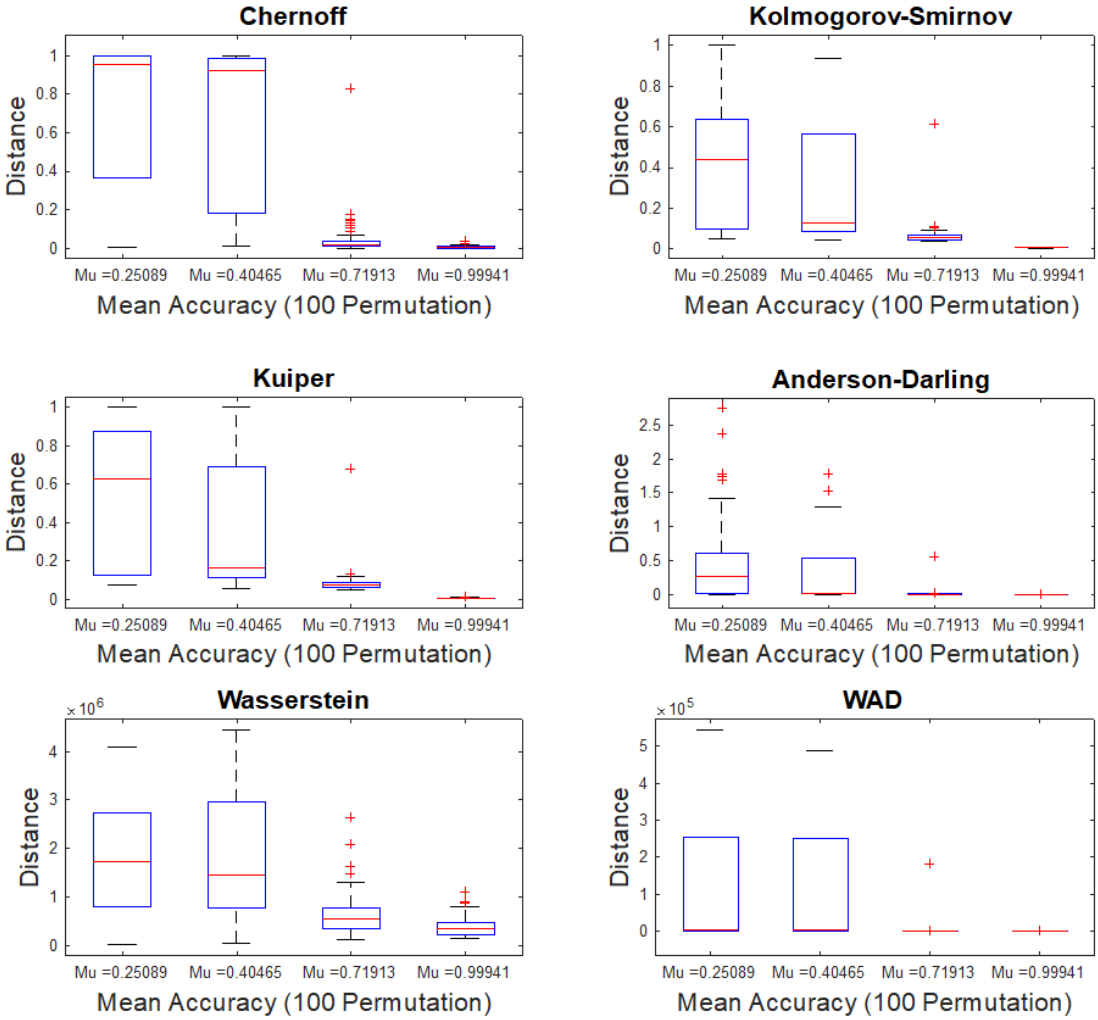


Figure 5-11. Box plot of statistical distance measures vs. accuracy over 100 iterations for Thursday Security Intrusion Detection in CICIDS2017 dataset

The rest of results for Security Intrusion Detection in CICIDS2017 dataset are available in the SafeML GitHub Repository (https://github.com/ISorokos/SafeML).

Figure 5-12 shows Pearson's correlation between the classes of Wednesday's data and the statistical ECDF-based distances. As can be seen, the WD and WAD distances have more correlation with the classes. This figure also shows the correlation between the measures themselves. The KSD and KD appear to be correlated. The WD and WAS also seemed to be correlated. These correlations can be explained due to the similarity in their formulation. P-values for the above correlations were evaluated to be zero, thereby validating the correlation hypotheses above.

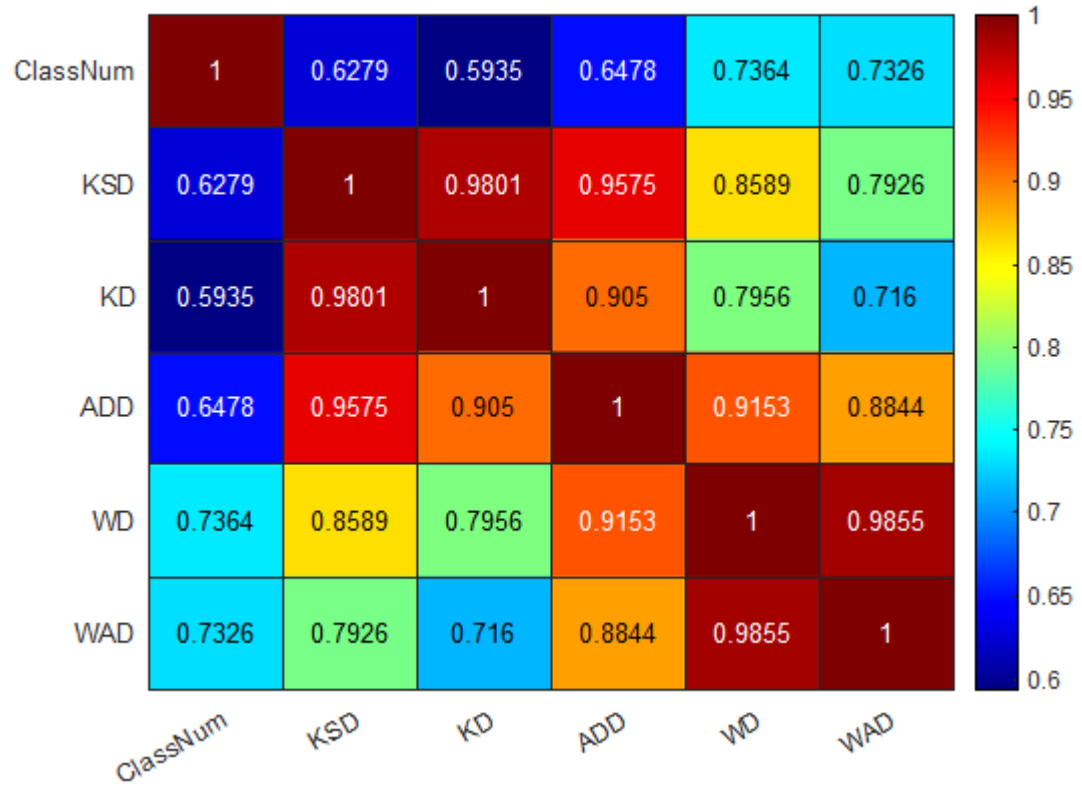


Figure 5-12. Correlation between class label numbers and statistical ECDF-based Distance Measures

5.5.3 Example 3: German Traffic Sign Recognition Benchmark (GTSRB)

In this section, numerical results comparing the proposed approach and existing approaches in the literature are presented for a German Traffic Sign Recognition (GTSR) dataset. The dataset has been released in 2011 and it includes 43 different traffic signs. The dataset is unbalanced and the number of samples for some classes can be more than the others. Regarding the cross-validation, the hold-out method is used to split 80\% of the data for training and 20\% for validation. It should be noted that the dataset has a separate folder for test data.

As mentioned before, the SafeML is a model-agnostic approach that can be used on top of any machine learning classifier regardless of its structure. In this paper, a Deep Convolutional Neural Network (CNN) classifier is used because of its reputation on image classification. The following structure is used as the configuration of CNN. The input has a 2D convolution layer (Conv2D) with a filter size of 32, kernel size of 5x5 and the Relu activation function. The second layer has another Conv2D with a filter size of 64, kernel size 3x3 and the Relu activation function. Then, a max pooling layer with size of 2x2 and a dropout layer with a rate 0.25 is used. After that, another Conv2D layer with a filter size of 64, kernel size of 3x3 and Relu activation function is added. A max pooling with the size of 2x2 and a dropout with the rate of 0.25 is applied on top of it. A flatten

and dense layer with size of 256, and Relu activation function, with 0.5 percent dropout is used. Finally, for the output, a dense layer with the size of 43 and Softmax activation function is considered. Moreover, the Adaptive Moment Estimation (ADAM) optimiser, and the cross-entropy loss function are used in the training procedure.

Using the above configuration, the performance of the CNN classifier was 0.9797 on the test dataset. The next level is to check whether the achieved accuracy is high enough or not? This part was not considered in the first version of the SafeML, and it could reduce the precision of the proposed approach when a poor classifier is chosen in the offline phase. In the case of having a poor classifier, the loop should be repeated until reaching a certain level of satisfaction for the accuracy. It is also possible to consider explainability approaches to make sure the trained classifier behaves reasonably and focuses on the right part of the image.

Assuming that the level of achieved accuracy is acceptable for safety experts, the images of each class will be separated to R, G, and B matrix and converted to the flatten vectors accordingly. As the size of each image is 30x30, the equivalent vector will be 1x900. The ECDFs of each class will be generated and stored for use in the next phase. In the online phase, the buffer size is considered as 15. In a practical scenario, the buffer size should be defined by safety experts and designers. As there was no real-time data, the test data are considered as the upcoming data, and we are going to see how the proposed approach will react to the wrong decisions.

To have better visualization, class number three is chosen. This class is related to the 60 Km speed limit sign, and it has 1410 images in the training dataset and 450 images in the test dataset. Various risks can be considered for miss-classification of this sign like having a lower speed and blocking the road or having a higher speed and increasing the probability of hitting pedestrians passing the street. The associated risk for miss-classification of each class can be investigated in a separate research study. The accuracy of the classifier for this class specifically was 0.9655. In other words, 435 images are detected correctly but 15 images are detected as the other classes. The following Sankey plot illustrates the relation between true labels and the predicted labels. As can be seen in this plot, the dataset is imbalanced and there some classes with low number of samples and other classes with a higher number of samples. In addition, it is clear that most of the images has been correctly classified.

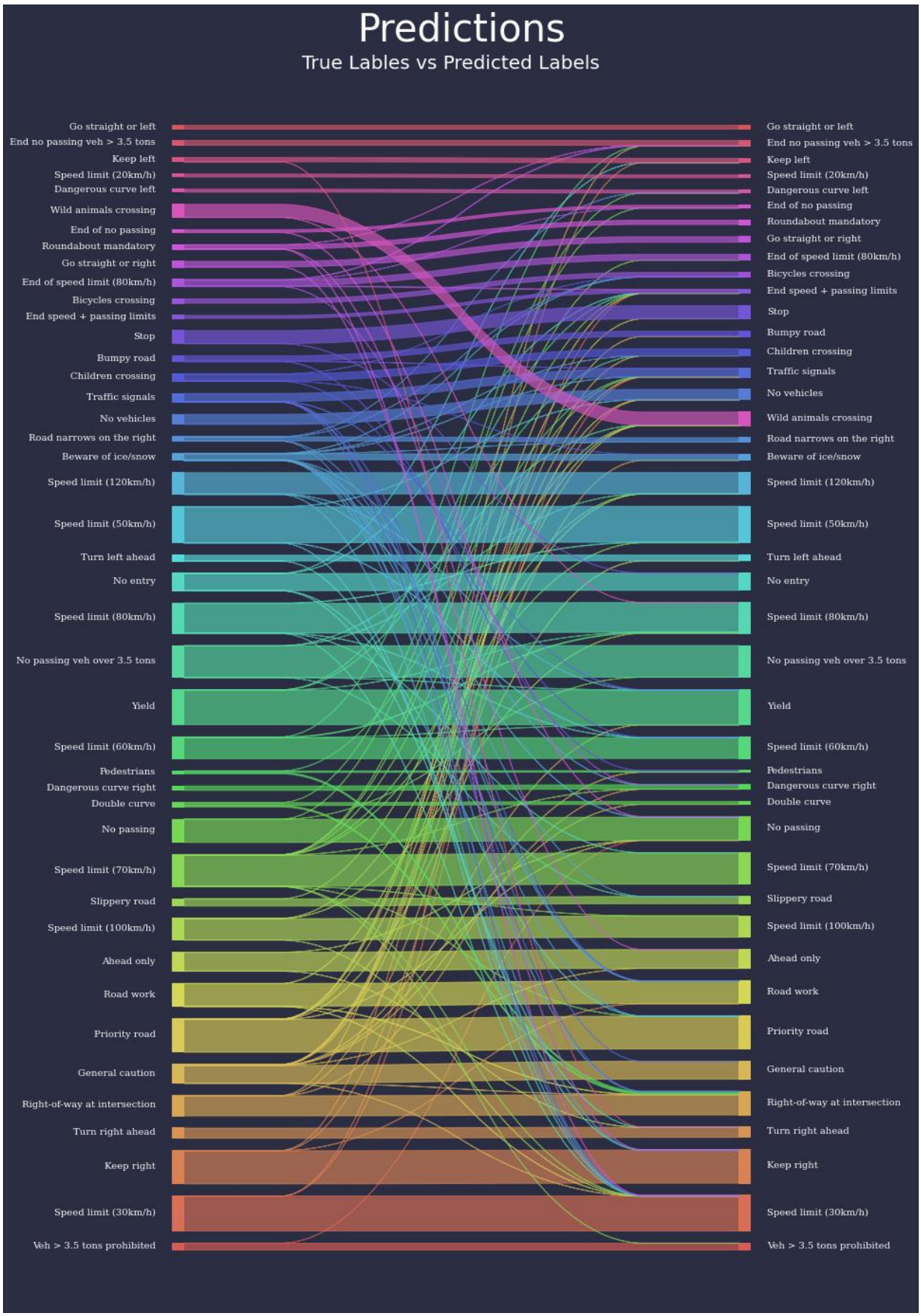


Figure 5-13. Sankey Plot showing the relation between true labels and the predicted labels.

Based on SafeML procedure, the R, G, B matrix of test images are converted to flatten vectors and their ECDFs have been generated. Furthermore, using the ECDF-based statistical distance measures such as Kolmogorov-Smirnov (KS), Kuiper (K), Anderson-Darling (AD), Cramer-Von Mises (CVM), and Wasserstein (W), the statistical distances will be obtained. The first version of SafeML will jump to a comparison between statistical distance measures and the pre-defined expected confidence threshold. However, in SafeML, a bootstrap algorithm with 1000 iterations is used to obtain the P-value and validate the measures (Gilleland, 2020). Thus, the measures with P-value lower than 0.05 are stored and others will be omitted. The validated statistical distance measure can be compared with the expected confidence level.

It should be noted that for each ECDF-based statistical distance measure, there should be a particular expected confidence threshold predefined by a safety expert. The decision of the machine learning classifier is accepted and trusted if the distance measure is higher than the predefined threshold. Additionally, a report of the statistical distance measure will be stored in a database to be used for the further development of the system.

In the situation that the statistical distance measure is 5% lower than the predefined threshold, the system may ask for further data. It should also be mentioned that in that situation, the autonomous vehicle can use other existing sources of information to validate the decision. For example, the autonomous vehicle can communicate with nearby vehicles or use the GPS and pre-loaded map data. The mentioned percentage can also be changed based on the safety experts' and system designers' opinion. At the moment there is no published standard to define these levels but, in the future, these parameters can be defined using the published standards.

The worst scenario is that the statistical distance measure is hugely different from the expected threshold, meaning the upcoming data has not been seen by the classifier before and there is a risk of missed classification. The SafeML idea is to put human-in-the-loop and ask the driver to make the decision. It is assumed that the driver has enough time for making the decision. However, there might be some cases where the time is restricted and SafeML cannot be used. As mentioned before, the autonomous vehicles that do not have wheel-based driving capability, it is suggested that a human agent from the control centre control the car remotely.

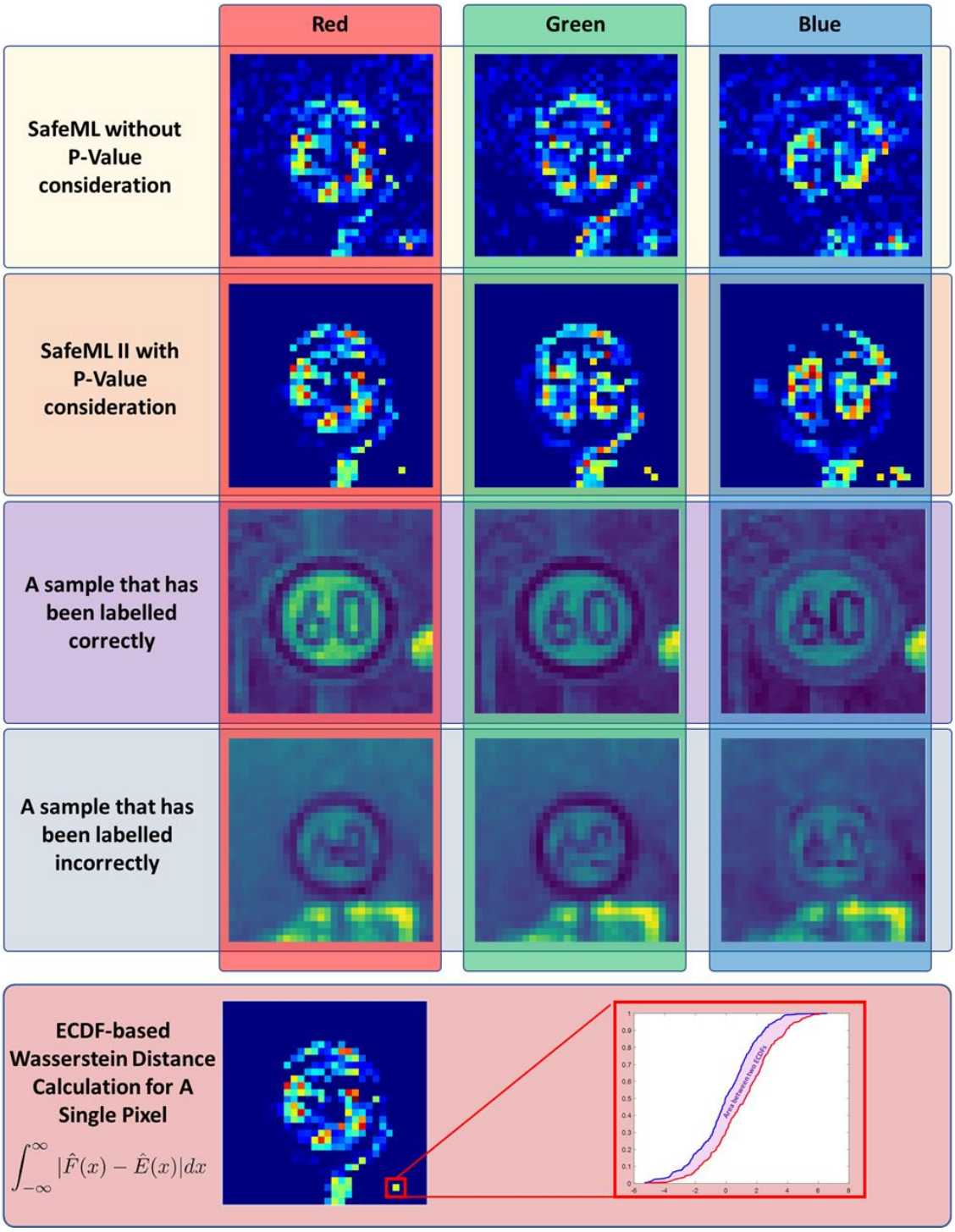


Figure 5-14. Sample results of SafeML with Wasserstein Dist. and considering p-values (class number 3)

The first row of the following figure illustrates the Wasserstein distance measure of the 60 Km traffic sign (Class 3) for R, G, and B part of the images. As can be seen, the middle of the image has more statistical differences in all three colour layers. Besides, the blue part of the image has less statistical distance in comparison to the red and green parts of the image.

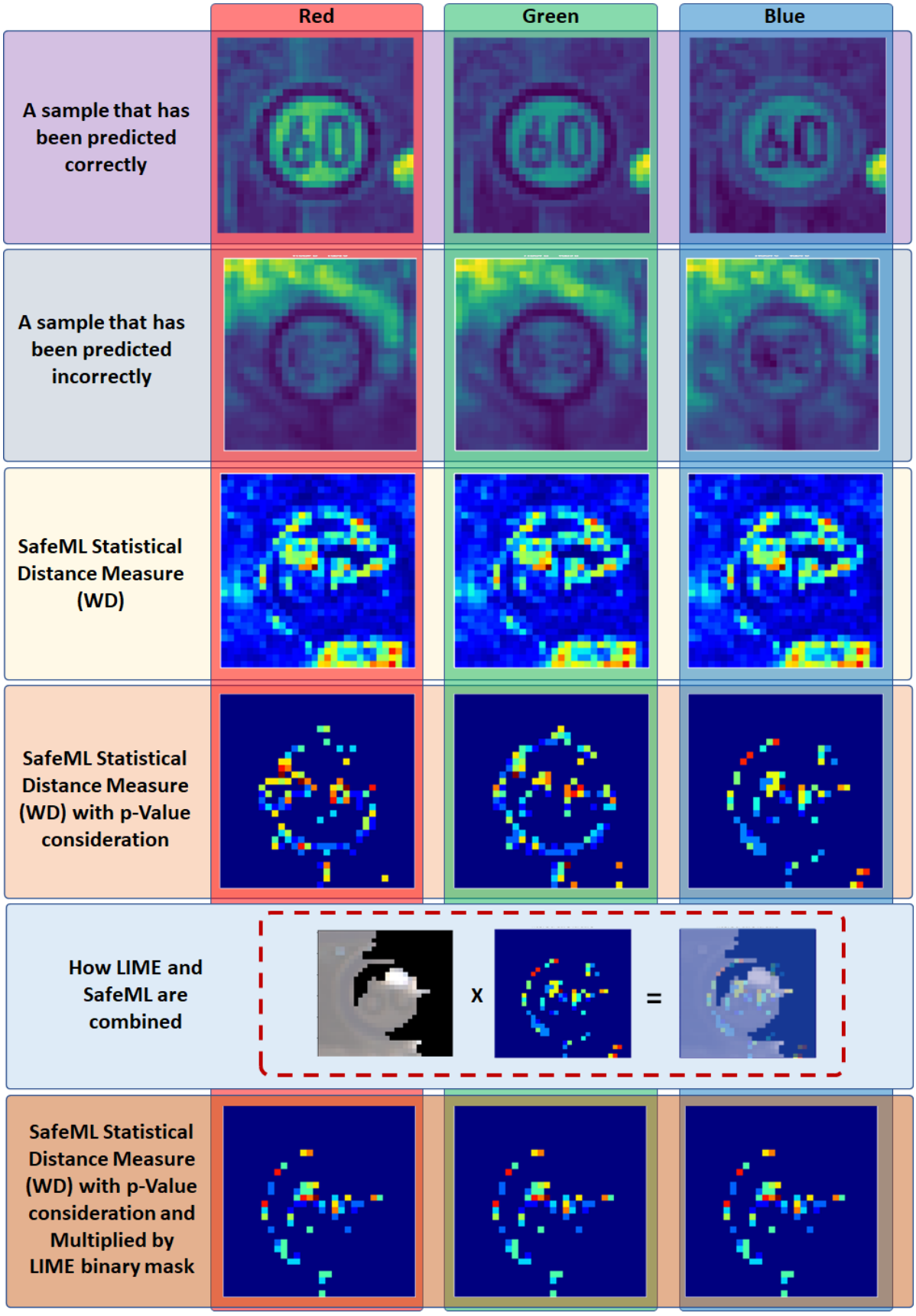


Figure 5-15. Sample result on classifier explainability on German Traffic Sign Recognition Benchmark using combined SafeML and LIME

As can be seen, in the first layer, a previous version of SafeML is used which has lack of P-value based distance validation while in the second row the SafeML is used that has the embedded P-value distance validation. Comparing the first and second row of this figure, it is clear that SafeML has a better statistical distance representation, and it does not catch the background areas of the signs.

The third row of this figure illustrates a sample image where the classifier has correctly detected the sign, while the fourth row shows a sample image where the classifier was not able to detect the sign correctly. However, it seems that it can be detected by a human with careful observation. Therefore, in these cases, human-in-the-loop can help the system to make the right decision and also learn it to make better decisions in the future. The AI system can be considered as a talented and clever child that needs to work in parallel with human and become mature over time. This figure also demonstrates how the ECDF-based Wasserstein is calculated for a pixel in the image.

The idea behind Figure 5-14 can be also extended by combining the idea of SafeML with explainability methods like LIME (short for local interpretable model-agnostic explanations) by (Ribeiro, Singh and Guestrin, 2016).

Figure 5-15 illustrates the idea of combining LIME with SafeML both for explainability and ML model monitoring. LIME is capable of finding top super-pixels responsible for getting a specific outcome in a ML model. For example, for 60 Km sign, it can show which super pixel are most influential one and create a mask based on them. The mask can then be multiplied with SafeML results.

In other words, it is like providing an additional weight for statistical distance measures. The statistical distance of the area that has higher explainability value will have more weight. On the other hand, the statistical distance of the area that has lower explainability value will have lower weight.

SafeML can provide estimated accuracy based on statistical distance measures. The following table provides a comparison between true accuracy, estimated accuracy by SafeML, and Wilson Interval Confidence bound from (Kläs and Sembach, 2019). For the Wilson Interval confidence, the z-score is chosen to be 3.29053 to gain 99.99\% confidence level. The Wilson interval confidence usually provides both upper bound and lower bound.

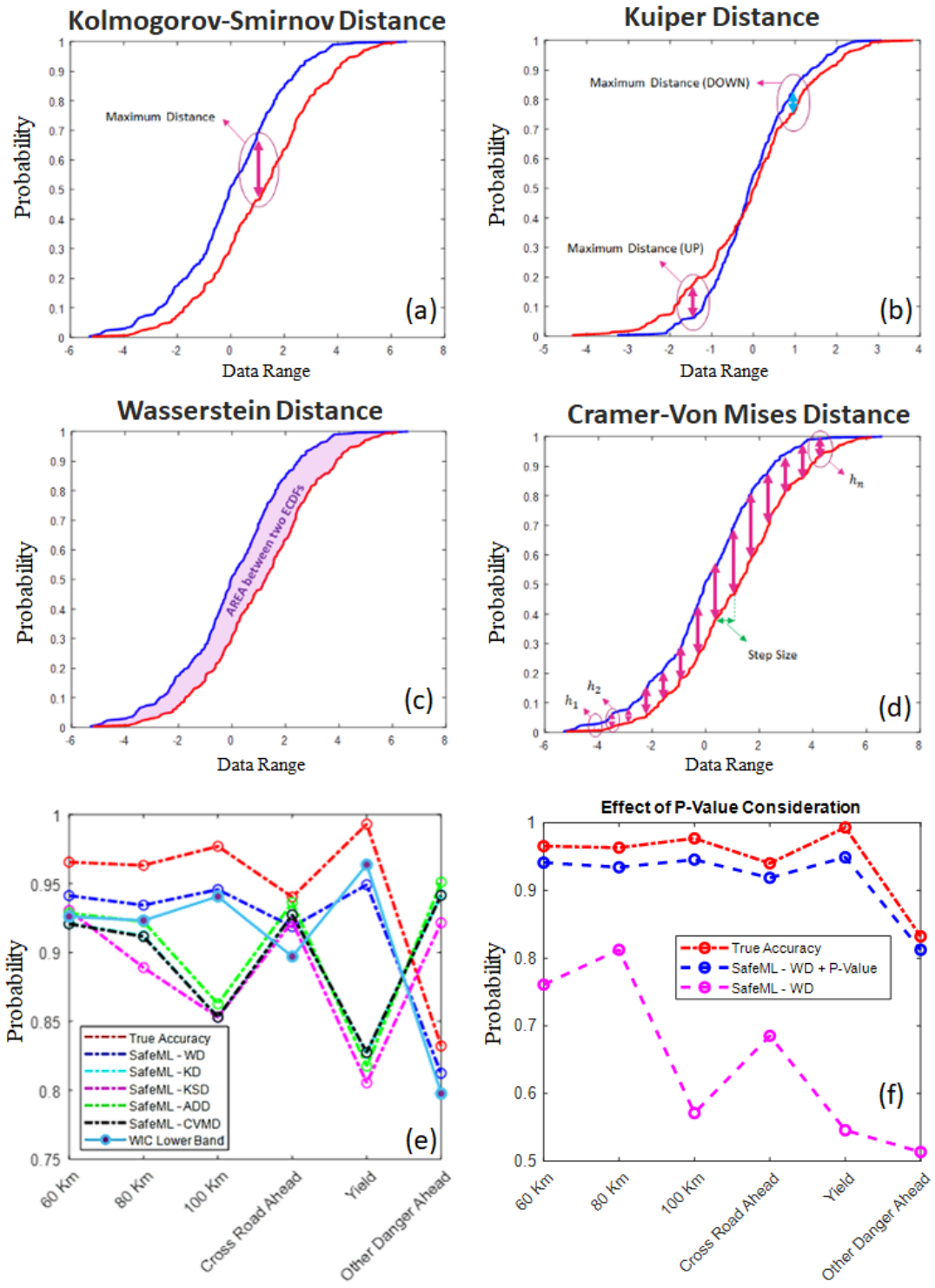
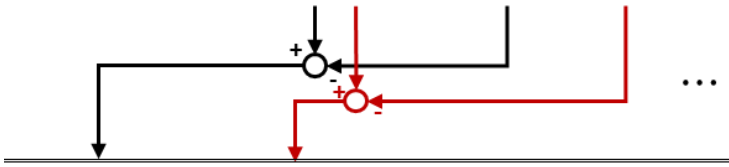


Figure 5-16. (a) Kolmogorov-Smirnov Distance, (b) Kuiper Distance, (c) Wasserstein Distance, (d) Cramer-Von Mises Distance, (e) Comparison between true accuracy, estimated accuracy by SafeML and (Kläs and Sembach, 2019), (f) Comparison between true accuracy, WD with and without P-Value consideration)

To ensure the maximum safety level, only the lower bound is considered. From the existing 43 classes in the GTSRB, 5 classes have been chosen for the comparison. The results show that in most cases the Wasserstein-based accuracy estimation has less error. For two cases the Wasserstein algorithm was not successful: for class number 11, the

Anderson-Darling estimation has less error and for class number 13, the low band Wilson Interval has better accuracy.

Class #	Sign Description	SafeML						Klas et al. 2019
		True Accuracy (TA)	KS	Kuiper	AD	CVM	Wasser stein	WIC Low
3	60 Km	0.9655	0.9305	0.9209	0.9285	0.9207	0.9412	0.9261
5	80 Km	0.9632	0.8889	0.9121	0.9223	0.9115	0.9343	0.9232
7	100 Km	0.9770	0.8530	0.8529	0.8627	0.8530	0.9456	0.9406
11	Cross Road Ahead	0.9402	0.9222	0.9274	0.9362	0.9273	0.9187	0.8970
13	Yield	0.9931	0.8053	0.8269	0.8174	0.8274	0.9491	0.9637
18	Other Danger Ahead	0.8321	0.9215	0.9419	0.9510	0.9414	0.8123	0.7976



SafeML					Klas et al. 2019
KS vs. TA	K vs. TA	AD vs. TA	CVM vs. TA	W vs. TA	WIC Low vs. TA
0.0350	0.0446	0.0370	0.0448	0.0243	0.0394
0.0743	0.0511	0.0409	0.0517	0.0289	0.0400
0.1240	0.1241	0.1143	0.1240	0.0314	0.0364
0.0180	0.0128	0.0040	0.0129	0.0215	0.0432
0.1878	0.1662	0.1757	0.1657	0.0440	0.0294
-0.0894	-0.1098	-0.1189	-0.1093	0.0198	0.0345

Figure 5-17. Comparison between true accuracy, estimated accuracy by SafeML II and Klas et al. 2019.

In this example, we have only focused on traffic sign recognition and the idea can be integrated with other safety-related parts of autonomous vehicle software to cover wider safety perspectives. For example, in (Kabir, Sorokos, *et al.*, 2019), it was explained how to build an integrated safety model and consider different components of a cooperative operation scenario of autonomous vehicles. The results of SafeML can be used as an input in the proposed safety model in that work to improve confidence in the provided assurance. It should be noted that the SafeML concept has some limitations. For example, it can only work with Machine Learning classifiers, while having the SafeML concept to work for prediction and regression algorithms is still an open research question. Moreover, we currently investigate what specific characteristics of a dataset can lead to a better ECDF-based statistical accuracy estimation in run time.

5.6 Capabilities and limitations

In this section we discuss the capabilities and limitations of SAFEML and make suggestions for further work to address limitations

5.6.1 Capabilities of the proposed approach

- Through modifying the existing statistical distance and error bound measures, the proposed method enables to estimate the accuracy bound of the trained ML algorithm in the field with no label on the incoming data.
- A human-in-loop procedure was proposed to enable or disable ML algorithms during operation.
- The proposed approach is easy to implement, and it can support a variety of distributions, both exponential and normal distribution families.

5.6.2 Limitations of the proposed approach and possible solutions

- The proposed algorithm is only tackling the safety evaluation problem in ML classification. However, expansion for clustering or dimension reduction since these problems can be addressed using statistical distance measures.
- There are some cases in which SafeML results are not correlated with the performance of the Machine Learning classifiers. For example, consider the univariate example in this chapter and its fixed threshold. If one holds the left distribution and moves the right distribution to right, the performance of the classification task will be improved. The further it goes, the higher performance it gets. While the main assumption of SafeML is that when there is a statistical distance measure, it is expected to have the performance dropped. There are certain ideas to think about this issue: a) in this example, although the performance is increasing, the results are not reliable, and the problem should have been defined to limit the distribution of each class. If it was a measured temperature, then pushing the distribution to further right means very high value and a potential to harm the system. b) The SafeML distance measures could also be defined in a more advanced manner. In other words, instead of measuring the statistical distance between each class for each feature, the SafeML could form an anchor-based distance measuring the statistical distance of each feature of each class I) to its own class, II) to other classes and III) to the local decision boundary.

- The proposed method will not be able to detect some types of adversarial attacks. For example, the one-pixel adversarial attack will not change the probability density function and the proposed method cannot detect it (Su, Vargas and Sakurai, 2019).

5.7 Conclusion

The multiple safety problem of AI and ML algorithms have been briefly discussed and the chapter then focused on the distributional shift in datasets. Research is currently investigating this issue and trying to make the existing ML classifiers more robust. Some work has focused on adding noise, perturbation, and rotation on data to make sure a classifier is robust to those changes. In this report, a different perspective to the problem was selected. Statistical approaches were used to estimate the accuracy of the classifier when there is no label, for example during operation. Positive results has been reported using Kolmogorov-Smirnov and other measures. However, the study still has a long journey to become mature, but this thesis is merely a starting point. Some of the key capabilities and limitations were highlighted.

Chapter 6

6 Conclusion

The motivation of this work was the observation that systems are becoming increasingly complex, intelligent, and open adaptive and this is alarming in a world that is so dependent on technology. The focus of this thesis was on addressing four aspects that complexity and intelligence pose for dependability: complex architecture and behaviour, effect of environmental factors, the autonomous and open nature of new systems and behaviour learned from data. In each of the four aspects of challenge, the state-of-the-art for the existing techniques was studied and new solutions have been proposed to overcome and deal with the challenge.

6.1 Summary of Contributions

This PhD thesis introduced a groundbreaking hierarchical method for assessing system reliability and availability, uniquely integrating dynamic fault tree analysis with

semi-Markov processes. At the core of this research was the innovative Hierarchical DFT (HDFT) approach, which forms the backbone of the thesis. Building upon this foundational model, the thesis further extends its scope and applicability. It delved into sophisticated extensions that enhance the HDFT's capacity to manage intricate system behaviours, efficiently address maintenance requirements, evaluate reliability in real-time, and adeptly handle systems exhibiting intelligent characteristics. These extensions not only augment the HDFT model but also significantly contribute to the broader field of system dependability evaluation, offering versatile and robust solutions for complex, dynamic environments. The rest of contributions to the state-of-the-art have been discussed below with respects to the four aspects of challenged addressed:

Complex architecture and behaviour: In chapter 2, the thesis proposed a novel hierarchical model-based reliability analysis technique that combines Dynamic Fault Tree and Semi-Markov Process. The method is able to divide a large system into several manageable sub-systems for the quantitative reliability evaluation. It can handle non-exponential failure behaviours. It improves earlier related work (Aslansefat and Latif-Shabgahi, 2020) using the idea of Markov path that can break-down Markov and Semi-Markov models. This can reduce the number of states and transitions that leads to lower computation complexity and reduce the issue of state explosion in Markov models**.**

Environmental factors: In chapter 3, the thesis proposed a novel solution for modelling environmental factors in dependability by defining a semi-Markov-based maintenance model called “Butterfly Maintenance Model (BMM)” and complex basic events in dynamic fault trees. These events not only consider the effect of external variables but also update the failure profile using data from monitoring systems and parameters like estimated remaining useful life (RUL). External variables such as weather, logistics, transportation, and crew availability were addressed in the context of offshore wind farms and availability modelling was provided as a use case for this model.

The method considers both traditional and data-driven availability analysis by combining both simple failure distribution and remaining useful life (RUL)-based transition in a single MRGP. This is the first time that MRGP is combined with real-time transitions. The proposed model enables users to evaluate both condition-based maintenance and reliability-centred maintenance. The integrated BMM model incorporates logistics delays, weather delays and a daily briefing and transfer delays. Consideration of these accessibility and delay factors will lead to a more realistic availability evaluation of the offshore wind farm.

In the Markov modelling theories, the issue of considering the time-based deadline for a transition has rarely been discussed. The proposed model uses deadline transitions to model shift time limits of the O&M for offshore wind turbines**.** Furthermore, imperfect maintenance can make the model and its evaluation more realistic. However, the literature lacks proper modelling for imperfect maintenance. The proposed model is not only able to consider imperfect maintenance but also able to use this option to model the life extension of the turbine. The approach can model non-exponential system behaviours using MRGP.

Although the MRGP has higher computation complexity, at the same time offers more power in modelling complex failure behaviours. Addionally, the embedded Markov model can be used with Markov processes in cases where only exponential failure and delay behaviours are considered. In other words, the computational complexity of the model is flexible and can be reduced via simplifying assumptions**.**

Open nature of systems: In chapter 4, the thesis proposed an approach to run-time dependability evaluation using a new instantiation of the recently proposed EDDI concept. The approach is novel and demonstrates run-time use of Faulkt Trees and Markov Models as executable EDDIs for dynamic dependability evaluation. A new symptom layer has been introduced which receives the outcome from the diagnostics and monitoring system and updates the reliability of the robot at runtime. To illustrate the approach, a power network inspection use case was considered to show how real-time reliability evaluation can be used to anticipate imminent failures and prevent accidents by recommending appropriate responses.

Behaviour learned from data: The thesis developed a novel approach called SafeML to address the Safety of Machine Learning systems. This is a challenging task because such systems have a black box architecture and lack a model that can used as a basis for analysis. The approach provides safety monitoring of machine learning classifiers against distributional shift using statistical distance measures. It focuses on the runtime evaluation of machine learning algorithms and provides statistical measures to constantly monitor the performance of ML that can increase our ability to detect the effect of distributional shifts on accuracy of reasoning.

A novel proposed human-in-loop procedure was also proposed to integrate SafeML in processes that employ ML algorithms. The process defines when to accept the reasoning of algorithms, when to request more data and when to remove confidence on

ML and seek support from operators. SafeML is easy to implement, and it can support a variety of distributions, both exponential and normal distribution families**.**

6.2 Limitations

Limitations of the work with respect to the four aspects of challenge examined in the thesis are presented below:

Complex architecture & behaviour: The results of this method are approximate for DFTs, with repeated events making the results more approximate or less precise. It should be noted that the proximity of the method would not diminish its effectiveness. Despite approximation, the results obtained through this method are more precise than the result of some other published research works (Yuge & Yanagi, 2008).

In the proposed method, the output of SEQ, OR, and POR gates with any complexity in their inputs will be precise. In the case of AND and PAND gates, when a complex event or a sub-tree is connected to the first input and a simple event connected to the second input, the output will be precise. The output will be approximate for other gates and any other situations for AND and PAND gates.

In some benchmarks in which the shared or sliding spares are used, the proposed method is not able to solve shared spare gates. It is suggested to replace shared spares and use the reshaping rules (see appendix). It should be noted that these gates are then modelled as a semi-Markov model and generalized for any kind of failures distribution function.

In this study, it is assumed that the fault tree has only coherent events and there would be no guarantee for a fault tree with non-coherent events.

There are still some gates like pSAND and SAND that cannot be modelled through the universal gate. We hope to improve this universal gate to consider pSAND and other gates in future research. It would be also possible to define new gates such as semi-PAND in the future.

The example of repairable DFT is just provided to show that there might be a possibility to improve the approach for repairable DFTs. However, at the moment, there is no guarantee that the approach works for any other example of repairable DFT. In fact, this example is just an insight into potential future works.

Environmental factors: The proposed model does not handle common cause failures. The BMM model of Chapter 3 can provide an exact value for local availability. However, combining the BMM model with the hierarchical dynamic fault tree solution

of Chapter 2 does not give exact value for the availability and the calculations will only approximate. This issue can be solved by improving the solution provided for hierarchical dynamic fault tree and its availability evaluation. The BMM model and the availability evaluation are highly dependent on the correctness of RUL estimation. Thus, having a poor estimation can lead the model to provide wrong values for availability.

Open nature of systems: Currently, the provided experimental result of the proposed EDDI is for a single robot. However, the proposed SafeDrones approach is capable of being considered for multi-robot applications. This might be achieved by including the robot collaboration aspects. For instance, the communication between two robots and the effect of a communication loss in one robot can affect the reliability of the others. The approach is dependent on the accuracy and reliability of monitoring and diagnosis unit.

Behaviour learned from data: SafeML is currently only tackling the safety evaluation problem of the machine-learning based classification. It can potentially be expanded for clustering, dimension reduction or any problem that can be evaluated through statistical difference. Furthermore, the proposed method will not be able to detect some types of adversarial attacks. For example, the one-pixel adversarial attack will not change the probability density function and the proposed method cannot detect it (Su, Vargas and Sakurai, 2019).

6.3 Future Work

Regarding future works, the following research directions and improvements are suggested suing the structure of the four aspects of challenge addressed in the thesis:

Complex architecture and behaviour: The result of proposed hierarchical dynamic fault tree analysis is sometimes approximate, and this issue can be improved by defining transformations rules like the one that has been proposed by algebraic solutions of DFT.

Using the Semi-Markov Process enables us to consider non-exponential failure distributions. However, there are existing non-exponential failure distributions that cannot be transformed using the Laplace Stieltjes theorem. One solution is to convert the distribution to a series of exponential distributions.

Environmental factors: The uncertainty and sensitivity analysis of the proposed butterfly maintenance model can be further established in the future using perturbation theory (Do Van, Barros and Bérenguer, 2008). Additionally, based on the idea mentioned

in (Gheraibia *et al.*, 2019), it will be possible to use artificial intelligence to update the availability model and make it more adaptive.

Open nature of systems: Expand the EDDI approach presented to enable safety-security co-engineering in SafeDrones and develop approaches for application in multi-robot systems that communicate data and together optimise responses to situations where dependability issues are raised.

Behaviour learned from data: SafeML supports machine learning classifiers, so one avenue of extension is supporting Machine Learning Regression and Prediction Algorithms. Other avenues of investigation include a) considering Recurrent Methods and extending SafeML to deal with Multivariate Time Series. For example, it would be worth exploring how SafeML can be used on 12-Lead ECG signals that has been used for heart disease classification. Currently, the SafeML idea in being extended for single variate time series in collaboration by Fraunhofer IESE (Akram *et al.*, 2022). b) Improving SafeML to support adaptive and online-learning algorithms where the model learns over time and tunes itself. c) When dealing with datasets that have many features, the SafeML performance is dropping. Dimensionality reduction algorithms have been used to deal with this issue. However, it is possible to use feature importance as part of the SafeML procedure. This idea has been briefly discussed in chapter 5, but more experiments require to show its validity.

The Idea of SafeML is not only usable for Model monitoring but can also provide ML explainability. This project is currently on-going, and the project name is SMILE: Statistical Mode-agnostic Interpretability with Local Explanations.

For complex ML/DL algorithms like YOLO and their complex tasks like image segmentation, the use of SafeML directly on input datasets might not be a good practice. However, using the last layer of the model or the latent space would be a perfect option to use SafeML (it can be model-specific version of SafeML). In addition, using domain representation or latent features might be a good approach to improve SafeML performance in these applications.

As a final word, this PhD thesis has made steps towards addressing the challenges of complexity and intelligence in the context of dependability evaluation. The presented work has demonstrated the importance of innovative approaches in research for real-world problems. It is almost impossible to deliver research without limitations. The limitations and future works in this thesis remind us that our journey for knowledge has

not finished and we need to continuously search for solutions and try to have a profound understanding of the challenges.

6.4 Publications and Their Relationship with Each Chapter

ID	Reference	Chapter	Challenge	Relation
1	Aslansefat, K., & Latif-Shabgahi, G. R. (2020). A hierarchical approach for dynamic fault trees solution through semi-Markov process. IEEE Transactions on Reliability, 69(3), 986-1003.	Chapter 2	Complexity	The Proposed Approach
2	Aslansefat, K., Kabir, S. Gheraibia, Y. & Papadopoulos, Y. (2020). Dynamic Fault Tree Analysis: State-of-the-Art in Modelling, Analysis and Tools, In Reliability Management and Engineering: Challenges and Future Trends, Taylor & Francis.	Chapter 2	Complexity	The Literature Review
3	Kabir, S., Aslansefat, K., Sorokos, I., Papadopoulos, Y., & Gheraibia, Y. (2019). A Conceptual Framework to Incorporate Complex Basic Events in HiP-HOPS. International Symposium on Model-Based Safety and Assessment, Lecture Notes in Computer Science 11842: 109-124, Springer, ISBN: 978-3-030-32871-9.	Chapter 2	Complexity	The idea of using Complex Basic Event
4	Aslansefat, K. & Papadopoulos, Y. (2019). DREAM: Data-driven Reliability-centred Evolutionary Automated Maintenance for Offshore Wind Farms, Global Offshore Wind 2019 – London.	Chapter 3	Complexity	The Proposed Butterfly Maintenance Model
5	Aslansefat, K. & Papadopoulos, Y. (2020). A Conceptual Framework for Data-driven Reliability-centred Evolutionary and Automated Maintenance of Offshore Wind Farms. EERA DeepWind'2020, Norway.	Chapter 3	Complexity	The Proposed Butterfly Maintenance Model
6	Aslansefat, K., Nikolaou, P., Walker, M., Akram, M. N., Sorokos, I., Reich, J., ... & Papadopoulos, Y. (2022). SafeDrones: Real-Time Reliability Evaluation of UAVs using Executable Digital Dependable Identities. In Model-Based Safety and Assessment.	Chapter 4	Complexity	SafeDrones and Its Application

	IMBSA 2022. Lecture Notes in Computer Science, Springer.			
7	Aslansefat, K., Kabir, S., Abdullatif, A., Vasudevan, V., & Papadopoulos, Y. (2021). Toward improving confidence in autonomous vehicle software: A study on traffic sign recognition systems. IEEE Computer, 54(8), 66-76.	Chapter 5	Intelligence	SafeML for Image Data

During the PhD, I had research collaboration with other groups and the following papers can be listed:

ID	Reference	Note
1	Kabir, S., Sorokos, I., Aslansefat, K., Papadopoulos, Y., Gheraibia, Y., Wei, R. (2019). A Runtime Safety Analysis Concept for Open Adaptive Systems. International Symp. on Model-Based Safety Assessment, Lecture Notes in Computer Science, Springer.	Considering the challenge of being OpenAdaptive in modern systems and discussing a way to update the reliability profile for such systems.
2	Aslansefat, K., Marques, F., Mendonça, R., & Barata, J. (2019). A markov process-based approach for reliability evaluation of the propulsion system in multi-rotor drones. In Conference on Computing, Electrical and Industrial Systems, Springer.	The Markov models of Propulsion System in UAVs in SafeDrones are utilized from this paper.
3	Akram, M. N., Ambekar, A., Sorokos, I., Aslansefat, K., & Schneider, D. (2022). StaDRe and StaDRo: Reliability and Robustness Estimation of ML-based Forecasting using Statistical Distance Measures. arXiv preprint arXiv:2206.11116.	Expanding SafeML: 1) for Time series, 2) for Robustness Evaluation 3) for Regression Tasks
4	Farhad, A. H., Sorokos, I., Schmidt, A., Akram, M. N., Aslansefat, K., & Schneider, D. (2022). Keep your Distance: Determining Sampling and Distance Thresholds in Machine Learning Monitoring. In Model-Based Safety and Assessment. IMBSA 2022. Lecture Notes in Computer Science, Springer.	Improving SafeML: 1) Discussing how its threshold can be defined automatically. 2) Including more examples using SafeML inside Carla Simulator.
5	Walker, C., Rothon, C., Aslansefat, K., Papadopoulos, Y., & Dethlefs, N. (2022). A Deep Learning Framework for Wind Turbine Repair Action Prediction Using Alarm Sequences and Long Short-Term Memory Algorithms. In Model-Based Safety and Assessment. IMBSA 2022. Lecture Notes in Computer Science, Springer.	This was part of our collaboration with EDF Energy R&D in UK.
6	Gheraibia, Y., Kabir, S., Aslansefat, K., Sorokos, I., & Papadopoulos, Y. (2019). Safety+ AI: A Novel Approach to Update Safety Models Using Artificial Intelligence. IEEE Access, 7(1), 135855-135869.	In the SafeDrones section, the model repair part of the procedure is addressed in this paper.

7	Kabir, S., Aslansefat, K., Gope, P., Campean, F., & Papadopoulos, Y. (2022, August). Combining Drone-based Monitoring and Machine Learning for Online Reliability Evaluation of Wind Turbines. In 2022 International IEEE Conference on Computing, Electronics & Communications Engineering (iCCECE).	In this paper the combination of SafeML with Fault Tree for Real-Time Reliability Evaluation of Offshore Wind Turbine Blades is Discussed.

References

Abeynayake, G. *et al.* (2021) 'Analytical Model for Availability Assessment of Large-Scale Offshore Wind Farms including Their Collector System', *IEEE Transactions on Sustainable Energy*. Institute of Electrical and Electronics Engineers Inc., 12(4), pp. 1974–1983. doi: 10.1109/TSTE.2021.3075182.

Agrell, C. *et al.* (2018) 'Pitfalls of machine learning for tail events in high risk environments', in *Safety and Reliability – Safe Societies in a Changing World*. CRC Press, pp. 3043–3051. doi: 10.1201/9781351174664-381.

Aizpurua, J. I. *et al.* (2017) 'Supporting group maintenance through prognostics-enhanced dynamic dependability prediction', *Reliability Engineering \& System Safety*. Elsevier, 168, pp. 171–188.

Akram, M. N. *et al.* (2022) 'StaDRe and StaDRo: Reliability and Robustness Estimation of ML-Based Forecasting Using Statistical Distance Measures', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Science and Business Media Deutschland GmbH, 13415 LNCS, pp. 289–301. doi: 10.1007/978-3-031-14862-0_21/TABLES/2.

Alexander, R. *et al.* (2020) 'Safety Assurance Objectives for Autonomous Systems'. Safety Critical Systems Club.

Amini, A. *et al.* (2020) 'Deep evidential regression', *Advances in Neural Information Processing Systems*, 33.

Amodei, D. *et al.* (2016) 'Concrete Problems in AI Safety'. Available at: http://arxiv.org/abs/1606.06565 (Accessed: 10 March 2020).

Andrews, J. and Tolo, S. (2023) 'Dynamic and dependent tree theory (D2T2): A framework for the analysis of fault trees with dependent basic events', *Reliability Engineering & System Safety*. Elsevier, 230, p. 108959. doi: 10.1016/J.RESS.2022.108959.

Armengaud, E., Schneider, D., *et al.* (2021) 'DDI: A novel technology and innovation model for dependable, collaborative and autonomous systems', *Proceedings - Design, Automation and Test in Europe, DATE*. Institute of Electrical and Electronics Engineers Inc., 2021-February, pp. 1626–1631. doi: 10.23919/DATE51398.2021.9474049.

Armengaud, E., Schneider, D, *et al.* (2021) 'DDI: A novel technology and innovation model for dependable, collaborative and autonomous systems', in *2021*

Design, Automation \& Test in Europe Conference \& Exhibition (DATE), pp. 1626–1631.

Arnold, F., Belinfante, A., der Berg, F., *et al.* (2013) 'DFTCalc: a tool for efficient fault tree analysis', in *International Conference on Computer Safety, Reliability, and Security*, pp. 293–301.

Arnold, F., Belinfante, A., Van Der Berg, F., *et al.* (2013) *DFTCalc: a tool for efficient fault tree analysis (extended version)*.

Askarian, M. *et al.* (2016) 'Fusion of micro-macro data for fault diagnosis of a sweetening unit using Bayesian network', *Chemical Engineering Research and Design*. Elsevier, 115, pp. 325–334. doi: 10.1016/J.CHERD.2016.09.008.

Aslansefat, K. (2014) 'A novel approach for reliability and safety evaluation of control systems with dynamic fault tree', *MSc. Thesis, Shahid Beheshti University*.

Aslansefat, K., Kabir, S., *et al.* (2020) 'Dynamic Fault Tree Analysis: State-of-the-Art in Modeling, Analysis, and Tools', in *Reliability Management and Engineering: Challenges and Future Trends*. CRC Press- Taylor \& Francis, pp. 73–111.

Aslansefat, K., Bahar Gogani, M., *et al.* (2020) 'Performance evaluation and design for variable threshold alarm systems through semi-Markov process', *ISA Transactions*, 97(1), pp. 282–295. doi: 10.1016/j.isatra.2019.08.015.

Aslansefat, K., Sorokos, I., *et al.* (2020) 'SafeML: Safety Monitoring of Machine Learning Classifiers through Statistical Difference Measure'. Available at: http://arxiv.org/abs/2005.13166 (Accessed: 30 May 2020).

Aslansefat, K. *et al.* (2021) 'Towards Improving Confidence in Autonomous Vehicle Software: A Study on Traffic Sign Recognition Systems', *Computer*. IEEE, 54(8), pp. 66–76. doi: 10.1109/MC.2021.3075054.

Aslansefat, K. *et al.* (2022) 'SafeDrones: Real-Time Reliability Evaluation of UAVs Using Executable Digital Dependable Identities', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Science and Business Media Deutschland GmbH, 13525 LNCS, pp. 252–266. doi: 10.1007/978-3-031-15842-1_18/FIGURES/5.

Aslansefat, K. and Latif-Shabgahi, G. R. (2020) 'A Hierarchical Approach for Dynamic Fault Trees Solution through Semi-Markov Process', *IEEE Transactions on Reliability*, 69(3), pp. 986–1003. doi: 10.1109/TR.2019.2923893.

Aslansefat, K. and Papadopoulos, Y. (2019) 'DREAM: Data-driven Reliability-centred Evolutionary Automated Maintenance for Offshore Wind Farms', in *Global*

Offshore Wind, London, UK.

Aslansefat, K. and Papadopoulos, Y. (2020) 'A Conceptual Framework for Data-driven Reliability-centred Evolutionary and Automated Maintenance of Offshore Wind Farms', in *EERA DeepWind'2020, Norway.*

Avizienis, A. *et al.* (2004) 'Basic concepts and taxonomy of dependable and secure computing', *IEEE Transactions on Dependable and Secure Computing*, 1(1), pp. 11–33. doi: 10.1109/TDSC.2004.2.

Bavuso, S. J. *et al.* (1994) 'HiRel: Hybrid Automated Reliability Predictor (HARP) Integrated Reliability Tool System (Version 7.0)', *NASA Technical Paper*. Citeseer, 3452, pp. 1–4.

Bellemare, M. G. *et al.* (2017) 'The Cramer Distance as a Solution to Biased Wasserstein Gradients'. Available at: http://arxiv.org/abs/1705.10743 (Accessed: 13 March 2020).

Boudali, H., Crouzen, P. and Stoelinga, M. (2007) 'Dynamic fault tree analysis using input/output interactive Markov chains', in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, pp. 708–717.

Boudali, H., Crouzen, P. and Stoelinga, M. (2010) 'A rigorous, compositional, and extensible framework for dynamic fault tree analysis', *IEEE Transactions on Dependable and Secure Computing*. IEEE, 7(2), pp. 128–143.

Boyd, M. A. (1992) *Dynamic fault tree models: techniques for analysis of advanced fault tolerant computer systems*. Duke University.

Brameret, P.-A., Rauzy, A. and Roussel, J.-M. (2015) 'Automated generation of partial Markov chain from high level descriptions', *Reliability Engineering \& System Safety*. Elsevier, 139, pp. 179–187.

Cai, B. *et al.* (2015) 'Real-time reliability evaluation methodology based on dynamic Bayesian networks: A case study of a subsea pipe ram BOP system', *ISA Transactions*. Elsevier, 58, pp. 595–604. doi: 10.1016/J.ISATRA.2015.06.011.

Chaux, P.-Y. *et al.* (2013) 'Towards a unified definition of Minimal Cut Sequences', *IFAC Proceedings Volumes*, 46(22), pp. 1–6. doi: https://doi.org/10.3182/20130904-3-UK-4041.00013.

Chen, G. and Ge, Z. (2018) 'Hierarchical Bayesian network modeling framework for large-scale process monitoring and decision making', *IEEE Transactions on Control Systems Technology*, pp. 1–9. doi: 10.1109/TCST.2018.2882562.

Cheng, C.-H., Huang, C.-H. and Nührenberg, G. (2019) 'nn-dependability-kit:

Engineering neural networks for safety-critical autonomous driving systems', in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6.

Cheng, J. *et al.* (2019) 'An Imitation medical diagnosis method of hydro-turbine generating unit based on Bayesian network', *Transactions of the Institute of Measurement and Control*. SAGE PublicationsSage UK: London, England, pp. 1–15. doi: 10.1177/0142331219826665.

Chiacchio, F. *et al.* (2013) 'A Weibull-based compositional approach for hierarchical dynamic fault trees', *Reliability Engineering and System Safety*. Elsevier Ltd, 109, pp. 45–52. doi: 10.1016/j.ress.2012.07.005.

Crouzen, P. (2006) *Compositional analysis of dynamic fault trees using input/output interactive markov chains*. University of Twente.

Distefano, S., Longo, F. and Trivedi, K. S. (2012a) 'Investigating dynamic reliability and availability through state-space models', *Computers and Mathematics with Applications*, 64(12), pp. 3701–3716. doi: 10.1016/j.camwa.2012.02.038.

Distefano, S., Longo, F. and Trivedi, K. S. (2012b) 'Investigating dynamic reliability and availability through state–space models', *Computers & Mathematics with Applications*. Pergamon, 64(12), pp. 3701–3716. doi: 10.1016/J.CAMWA.2012.02.038.

Dominguez-Garcia, A. D. *et al.* (2008) 'An integrated methodology for the dynamic performance and reliability evaluation of fault-tolerant systems', *Reliability Engineering \& System Safety*. Elsevier, 93(11), pp. 1628–1649.

Dubrova, E. (2013) 'Fault-tolerant design', *Fault-Tolerant Design*. Springer New York, pp. 1–185. doi: 10.1007/978-1-4614-2113-9/COVER.

Dugan, J. (1993) 'Analysis of a hardware and software fault tolerant processor for critical applications', in *9th Computing in Aerospace Conference*, p. 4573.

Dugan, J. B., Bavuso, S. J. and Boyd, M. A. (1992) 'Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems', *IEEE Transactions on Reliability*, 41(3), pp. 363–377. doi: 10.1109/24.159800.

Dugan, J. B., Bavuso, S. J. and Boyd, M. A. (1993) 'Fault trees and Markov models for reliability analysis of fault-tolerant digital systems', *Reliability Engineering \& System Safety*. Elsevier, 39(3), pp. 291–307.

Dugan, J. B., Sullivan, K. J. and Coppit, D. (2000) 'Developing a low-cost high-quality software tool for dynamic fault-tree analysis', *IEEE Transactions on Reliability*, 49(1), pp. 49–59. doi: 10.1109/24.855536.

Dugan, J. B., Venkataraman, B. and Gulati, R. (1997) 'DIFTree: a software package

for the analysis of dynamic fault tree models', in *Annual Reliability and Maintainability Symposium*, pp. 64–70.

Elderhalli, Y. *et al.* (2017) 'Dynamic Fault Trees Analysis using an Integration of Theorem Proving and Model Checking', *CoRR*, abs/1712.0.

Fussell, J. B., Aber, E. F. and Rahl, R. G. (1976) 'On the Quantitative Analysis of Priority-AND Failure Logic', *IEEE Transactions on Reliability*, R-25(5), pp. 324–326.

Garg, H. (2015) 'An approach for analyzing the reliability of industrial system using fuzzy kolmogorov's differential equations', *Arabian Journal for Science and Engineering*. Springer, 40(3), pp. 975–987.

Ge, D. *et al.* (2015) 'Quantitative analysis of dynamic fault trees using improved sequential binary decision diagrams', *Reliability Engineering \& System Safety*. Elsevier, 142, pp. 289–299.

Ge, D. *et al.* (2016) 'Quantification of highly coupled dynamic fault tree using IRVPM and SBDD', *Quality and Reliability Engineering International*. Wiley Online Library, 32(1), pp. 139–151.

Ge, D. and Yang, Y. (2015) 'Reliability analysis of non-repairable systems modeled by dynamic fault trees with priority AND gates', *Applied Stochastic Models in Business and Industry*. Wiley Online Library, 31(6), pp. 809–822.

Ge, D. and Yang, Y. (2017) 'Negating a generalized cut sequence: Bridging the gap between dynamic fault trees quantification and sum of disjoint products methods', *Quality and Reliability Engineering International*. Wiley Online Library, 33(2), pp. 357–367.

Gehr, T. *et al.* (2018) 'Ai2: Safety and robustness certification of neural networks with abstract interpretation', in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18.

Gerasimou, S. *et al.* (2020) 'Importance-driven deep learning system testing', in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pp. 702–713.

Getir, S. *et al.* (2018) 'Supporting semi-automatic co-evolution of architecture and fault tree models', *Journal of Systems and Software*. Elsevier, 142, pp. 115–135. doi: 10.1016/J.JSS.2018.04.001.

Gheraibia, Y. *et al.* (2019) 'Safety + AI: A Novel Approach to Update Safety Models Using Artificial Intelligence', *IEEE Access*, 7, pp. 135855–135869. doi: 10.1109/ACCESS.2019.2941566.

Gilleland, E. (2020) ‘Bootstrap methods for statistical inference. Part II: Extreme-value analysis’, *Journal of Atmospheric and Oceanic Technology*. American Meteorological Society, 37(11), pp. 2135–2144.

Goodfellow, I. J. *et al.* (2014) *Generative Adversarial Nets*. Available at: http://www.github.com/goodfeli/adversarial (Accessed: 13 March 2020).

Gulrajani, I. *et al.* (2017) ‘Improved training of wasserstein gans’, in *Advances in neural information processing systems*, pp. 5767–5777.

Guo, W. G., Han, W. and Liu, S. Y. (2011) ‘Dynamic Fault Tree Based on Weibull Distribution’, in *Advanced Materials Research*, pp. 1322–1327.

Hao, J., Zhang, L. and Wei, L. (2014) ‘Reliability analysis based on improved dynamic fault tree’, in *Engineering Asset Management 2011*. Springer, pp. 283–299.

Hermanns, H. (2002) ‘Interactive Markov Chains, volume 2428 of’, *Lecture Notes in Computer Science*.

Huang, C.-Y. and Chang, Y.-R. (2007) ‘An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees’, *Reliability Engineering \& System Safety*. Elsevier, 92(10), pp. 1403–1412.

Huang, H. *et al.* (2013) ‘Fuzzy dynamic fault tree analysis for the solar array drive assembly’, *Journal of Mechanical Engineering*, 49(19), pp. 70–76.

Huang, L. *et al.* (2016) ‘Modeling of operational availability of offshore wind turbines’, in *2016 International Conference on Probabilistic Methods Applied to Power Systems, PMAPS 2016 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/PMAPS.2016.7764123.

Huang, L. L. *et al.* (2017) ‘A Markov-Chain-Based Availability Model of Offshore Wind Turbine Considering Accessibility Problems’, *IEEE Transactions on Sustainable Energy*. Institute of Electrical and Electronics Engineers Inc., 8(4), pp. 1592–1600. doi: 10.1109/TSTE.2017.2695661.

Hurdle, E. E., Bartlett, L. M. and Andrews, J. D. (2007) ‘System fault diagnostics using fault tree analysis’, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. SAGE PublicationsSage UK: London, England, 221(1), pp. 43–55. doi: 10.1243/1748006XJRR6.

ISO (2017) ‘ISO/IEC JTC 1/SC 42: Artificial intelligence’. Available at: https://www.iso.org/committee/6794475.html.

Kabir, S. *et al.* (2017) ‘A model-based extension to HiP-HOPS for dynamic fault propagation studies’, in *International Symposium on Model-Based Safety and*

Assessment, pp. 163–178. doi: 10.1007/978-3-319-64119-5_11.

Kabir, S. (2017a) 'An overview of fault tree analysis and its application in model based dependability analysis', *Expert Systems with Applications*. Pergamon, 77, pp. 114–135. doi: 10.1016/J.ESWA.2017.01.058.

Kabir, S. (2017b) 'An overview of fault tree analysis and its application in model based dependability analysis', *Expert Systems with Applications*. Elsevier, 77, pp. 114–135.

Kabir, S., Aslansefat, K., *et al.* (2019) 'A Conceptual Framework to Incorporate Complex Basic Events in HiP-HOPS', pp. 109–124. doi: 10.1007/978-3-030-32872-6_8.

Kabir, S., Sorokos, I., *et al.* (2019) 'A Runtime Safety Analysis Concept for Open Adaptive Systems', pp. 332–346. doi: 10.1007/978-3-030-32872-6_22.

Kabir, S. *et al.* (2020) 'A Hybrid Modular Approach for Dynamic Fault Tree Analysis', *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., 8, pp. 97175–97188. doi: 10.1109/ACCESS.2020.2996643.

Kaiser, B., Gramlich, C. and Förster, M. (2007) 'State/event fault trees- A safety analysis model for software-controlled systems', *Reliability Engineering {\&} System Safety*, 92(11), pp. 1521–1537.

Katz, G. *et al.* (2019) 'The marabou framework for verification and analysis of deep neural networks', in *International Conference on Computer Aided Verification*, pp. 443–452.

Kim, D. S., Ghosh, R. and Trivedi, K. S. (2010) 'A Hierarchical Model for Reliability Analysis of Sensor Networks', in *2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing*, pp. 247–248.

Kläs, M. and Jöckel, L. (2020) 'A Framework for Building Uncertainty Wrappers for AI/ML-Based Data-Driven Components', in *International Conference on Computer Safety, Reliability, and Security*, pp. 315–327.

Kläs, M. and Sembach, L. (2019) 'Uncertainty wrappers for data-driven models', in *International Conference on Computer Safety, Reliability, and Security*, pp. 358–364.

Lampis, M. and Andrews, J. D. (2009) 'Bayesian belief networks for system fault diagnostics', *Quality and Reliability Engineering International*. John Wiley & Sons, Ltd, 25(4), pp. 409–426. doi: 10.1002/qre.978.

Laurens van der Maaten (2014) 'Accelerating t-SNE using Tree-Based Algorithms', *Journal of Machine Learning Research*, 15, pp. 3221–3245. doi: 10.1007/978-1-60761-580-4_8.

Lee, Kimin *et al.* (2017) ‘Training confidence-calibrated classifiers for detecting out-of-distribution samples’, *arXiv preprint arXiv:1711.09325*.

Li, Y. F. *et al.* (2012) ‘A new fault tree analysis method : fuzzy dynamic fault tree analysis’, *Eksploatacja i Niezawodnosc-Maintenance and Reliability*, 14(3), pp. 208–214.

Liu, D. *et al.* (2007) ‘Cut Sequence Set Generation for Fault Tree Analysis’, in *Embedded Software and Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 592–603.

Manian, R. *et al.* (1999) ‘Bridging the gap between systems and dynamic fault tree models’, in *Annual Reliability and Maintainability. Symposium. 1999 Proceedings (Cat. No. 99CH36283)*, pp. 105–111.

Manno, G. *et al.* (2014) ‘Conception of Repairable Dynamic Fault Trees and resolution by the use of RAATSS, a Matlab®toolbox based on the ATS formalism’, *Reliability Engineering \& System Safety*. Elsevier, 121, pp. 250–262.

de Medeiros, I. P. *et al.* (2014) ‘PHM-based Multi-UAV task assignment’, in *2014 IEEE International Systems Conference Proceedings*, pp. 42–49.

Merle, G. (2010) *Algebraic modelling of Dynamic Fault Trees, contribution to qualitative and quantitative analysis*. ENS CACHAN.

Minato, S. (2001) ‘Zero-suppressed BDDs and their applications’, *International Journal on Software Tools for Technology Transfer*. Springer, 3(2), pp. 156–170.

Misra, K. B. (2008) ‘Performability Engineering: An Essential Concept in the 21st Century’, *Handbook of Performability Engineering*. Springer London, pp. 1–12. doi: 10.1007/978-1-84800-131-2_1.

Mohseni, S. *et al.* (2023) ‘Taxonomy of Machine Learning Safety: A Survey and Primer’, *ACM Computing Surveys*. Association for Computing Machinery (ACM), 55(8), pp. 1–38. doi: 10.1145/3551385.

Niwas, R. and Garg, H. (2018) ‘An approach for analyzing the reliability and profit of an industrial system based on the cost free warranty policy’, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*. Springer, 40(5), p. 265.

Norberg, T. *et al.* (2009) ‘Added value in fault tree analyses’, in *20Safety, Reliability and Risk Analysis: Theory, Methods and Applications’, joint ESREL (European Safety and Reliability) and SRA-Europe (Society for Risk Analysis Europe) conference, Valencia, Spain, 22nd--25th September*, pp. 1041–1049.

Norberg, T., Rosén, L. and Lindhe, A. (2009) ‘Added value in fault tree analyses’.

Taylor & Francis Group.

on Standards in Public Life, U. K. C. (2020) ‘Artificial Intelligence and Public Standards’. Available at: https://www.gov.uk/government/publications/artificial-intelligence-and-public-standards-report.

Ottavi, M. *et al.* (2014) ‘Dependable multicore architectures at nanoscale: The view from europe’, *IEEE Design \& Test*. IEEE, 32(2), pp. 17–28.

Palshikar, G. K. (2002) ‘Temporal fault trees’, *Information and Software Technology*, 44(3), pp. 137–150.

Panigrahi, R. and Borah, S. (2018) ‘A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems’, *International Journal of Engineering \& Technology*, 7(3.24), pp. 479–482.

Paterson, C. *et al.* (2021) ‘DeepCert: Verification of Contextually Relevant Robustness for Neural Network Image Classifiers’, *arXiv preprint arXiv:2103.01629*.

Pietrantuono, R., Popov, P. and Russo, S. (2020) ‘Reliability assessment of service-based software under operational profile uncertainty’, *Reliability Engineering \& System Safety*. Elsevier, 204, p. 107193.

Piriou, P.-Y., Faure, J.-M. and Lesage, J.-J. (2017) ‘Generalized Boolean logic Driven Markov Processes: A powerful modeling framework for Model-Based Safety Analysis of dynamic repairable and reconfigurable systems’, *Reliability Engineering \& System Safety*. Elsevier, 163, pp. 57–68.

Piriou, P.-Y., Faure, J.-M. and Lesage, J.-J. (2019) ‘Finding the minimal cut sequences of dynamic, repairable, and reconfigurable systems from Generalized Boolean logic Driven Markov Process models’, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. SAGE Publications Sage UK: London, England, pp. 1–12. doi: 10.1177/1748006X19827128.

Ramezani, Z. *et al.* (2016) ‘Hierarchical steady-state availability evaluation of dynamic fault trees through equal Markov model’, in *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. doi: 10.1109/iraniancee.2016.7585822.

Ranjbar, A. H., Kiani, M. and Fahimi, B. (2011) ‘Dynamic Markov model for reliability evaluation of power electronic systems’, in *2011 International conference on power engineering, energy and electrical drives*, pp. 1–6.

Rauzy, A. B. (2011) ‘Sequence Algebra, Sequence Decision Diagrams and Dynamic Fault Trees’, *Reliability Engineering \& System Safety*, 96(7), pp. 785–792. doi: https://doi.org/10.1016/j.ress.2011.02.005.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) '" Why should i trust you?" Explaining the predictions of any classifier', in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Santos, F., Teixeira, Â. P. and Soares, C. G. (2015) 'Modelling and simulation of the operation and maintenance of offshore wind turbines', *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. SAGE Publications Ltd, 229(5), pp. 385–393. doi: 10.1177/1748006X15589209.

Savva, A. *et al.* (2021) 'ICARUS: Automatic Autonomous Power Infrastructure Inspection with UAVs', in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 918–926.

Schneider, D. *et al.* (2015a) 'WAP: digital dependability identities', in *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 324–329.

Schneider, D. *et al.* (2015b) 'WAP: Digital dependability identities', *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE Computer Society, pp. 324–329. doi: 10.1109/ISSRE.2015.7381825.

Schneider, D. and Trapp, M. (2013) 'Conditional Safety Certification of Open Adaptive Systems', *ACM Trans. Auton. Adapt. Syst.* New York, NY, USA: Association for Computing Machinery, 8(2). doi: 10.1145/2491465.2491467.

Sensoy, M., Kaplan, L. and Kandemir, M. (2018) 'Evidential deep learning to quantify classification uncertainty', *arXiv preprint arXiv:1806.01768*.

Sharafaldin, I., Lashkari, A. H. and Ghorbani, A. A. (2018) 'Toward generating a new intrusion detection dataset and intrusion traffic characterization.', in *International Conference on Information Systems Security and Privacy(ICISSP)*, pp. 108–116. doi: 10.5220/0006639801080116.

Sharvia, S. *et al.* (2016) 'Model-based dependability analysis: state-of-the-art, challenges, and future outlook', in *Software Quality Assurance*. Elsevier, pp. 251–278.

She, X. *et al.* (2020) 'SAFE-DNN: A Deep Neural Network With Spike Assisted Feature Extraction For Noise Robust Inference', in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Shi, D., Yang, B. and Quan, Q. (2016) 'Reliability analysis of multicopter configurations based on controllability theory', in *2016 35th Chinese Control Conference (CCC)*, pp. 6740–6745.

Sikorska, J. Z., Hodkiewicz, M. and Ma, L. (2011) 'Prognostic modelling options for remaining useful life estimation by industry', *Mechanical Systems and Signal*

Processing. Academic Press, 25(5), pp. 1803–1836. doi: 10.1016/J.YMSSP.2010.11.018.

Simen, E. *et al.* (2018) *AI + SAFETY: Safety implications for Artificial Intelligence*. Available at: https://ai-and-safety.dnvgl.com.

Singh, G. *et al.* (2018) 'Fast and Effective Robustness Certification', *Advances in Neural Information Processing Systems*, 31. Available at: http://safeai.ethz.ch/. (Accessed: 20 April 2023).

Sorokos, I. (2021) *Open Dependability Exchange metamodel, DEIS EU Project*. Github.

Sun, Y. *et al.* (2023) 'Failure analysis of floating offshore wind turbines considering working condition correlations', *Quality and Reliability Engineering International*. John Wiley & Sons, Ltd. doi: 10.1002/QRE.3334.

Tambon, F. *et al.* (2022) 'How to certify machine learning based safety-critical systems? A systematic literature review', *Automated Software Engineering 2022 29:2*. Springer, 29(2), pp. 1–74. doi: 10.1007/S10515-022-00337-X.

Tang, Z. and Dugan, J. B. (2004) 'Minimal cut set/sequence generation for dynamic fault trees', in *Annual Reliability and Maintainability Symposium*, pp. 207–213. doi: 10.1109/RAMS.2004.1285449.

Tannous, O., Xing, L. and Dugan, J. B. (2011) 'Reliability analysis of warm standby systems using sequential BDD', in *Proceedings of Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7.

Trivedi, K. S. and Bobbio, A. (2017) *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge: Cambridge University Press. doi: 10.1017/9781316163047.

Verma, A. K., Srividya, A. and Karanki, D. R. (2010) 'System reliability modeling', *Reliability and Safety Engineering*. Springer, pp. 71–168.

Vesely, W. *et al.* (2002) *Fault Tree Handbook with Aerospace Applications*. Washington, DC.

Volk, M., Junges, S. and Katoen, J.-P. (2018) 'Fast dynamic fault tree analysis by model checking techniques', *IEEE Transactions on Industrial Informatics*. IEEE, 14(1), pp. 370–379.

Walker, M. (2009) *Pandora: A Logic for the Qualitative Analysis of Temporal Fault Trees*. University of Hull.

Wei, R. *et al.* (2017) 'Deis: Dependability engineering innovation for cyber-physical systems', in *Federation of International Conferences on software technologies:*

applications and foundations, pp. 409–416.

Wei, R. *et al.* (2018) 'DEIS: Dependability Engineering Innovation for Cyber-Physical Systems', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, 10748 LNCS, pp. 409–416. doi: 10.1007/978-3-319-74730-9_37/FIGURES/2.

Xiang, J. *et al.* (2012) 'Combinatorial Analysis of Dynamic Fault Trees with Priority-AND Gates', in *23rd International Symposium on Software Reliability Engineering Workshops*, pp. 3–4. doi: 10.1109/ISSREW.2012.27.

Xiang, J. *et al.* (2013) 'A Static Analysis of Dynamic Fault Trees with Priority-AND Gates', in *Sixth Latin-American Symposium on Dependable Computing*, pp. 58–67. doi: 10.1109/LADC.2013.14.

Xing, L., Shrestha, A. and Dai, Y. (2011) 'Exact combinatorial reliability analysis of dynamic systems with sequence-dependent failures', *Reliability Engineering \& System Safety*. Elsevier, 96(10), pp. 1375–1385.

Xing, L., Tannous, O. and Dugan, J. B. (2012) 'Reliability Analysis of Nonrepairable Cold-Standby Systems Using Sequential Binary Decision Diagrams', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(3), pp. 715–726. doi: 10.1109/TSMCA.2011.2170415.

Yan, L. *et al.* (2021) 'Reliability analysis of station autonomous computer system based on fuzzy dynamic fault tree and Markov model', *Engineering Reports*. John Wiley & Sons, Ltd, 3(8), p. e12376. doi: 10.1002/ENG2.12376.

Yevkin, O. (2016) 'An efficient approximate Markov chain method in dynamic fault tree analysis', *Quality and Reliability Engineering International*. Wiley Online Library, 32(4), pp. 1509–1520.

Yin, L., Smith, M. A. J. and Trivedi, K. S. (2001) 'Uncertainty analysis in reliability modeling', in *Annual Reliability and Maintainability Symposium. 2001 Proceedings. International Symposium on Product Quality and Integrity (Cat. No. 01CH37179)*, pp. 229–234.

Yuge, T. and Yanagi, S. (2008) 'Quantitative analysis of a fault tree with priority AND gates', *Reliability Engineering \& System Safety*. Elsevier, 93(11), pp. 1577–1583.

Zhang, H. *et al.* (2011) 'A method of quantitative analysis for dynamic fault tree', in *Proceedings of Annual Reliability and Maintainability Symposium*, pp. 1–6. doi: 10.1109/RAMS.2011.5754471.

Zhao, X. *et al.* (2021) 'Assessing the Reliability of Deep Learning Classifiers

Through Robustness Evaluation and Operational Profiles', *arXiv preprint arXiv:2106.01258*.

Zhu, D. *et al.* (2023) 'Probabilistic modeling for long-term fatigue reliability of wind turbines based on Markov model and subset simulation', *International Journal of Fatigue*. Elsevier, 173, p. 107685. doi: 10.1016/J.IJFATIGUE.2023.107685.

Zixian, L. *et al.* (2011) 'Gastric esophageal surgery risk analysis with a fault tree and Markov integrated model', *Reliability Engineering \& System Safety*. Elsevier, 96(12), pp. 1591–1600.